

SYNGRESS®

4 FREE BOOKLETS

YOUR SOLUTIONS MEMBERSHIP



Windows Linux Migration Toolkit

Your Windows to Linux Extreme Makeover!

- Fully Functioning Scripts on CD-ROM Automate Your Migration Tasks
- Complete Coverage of Migration Process Planning, Anti-Virus and Anti-Spam Applications, and Deployment Details
- Covers Windows 95; 98, 98SE, and Me; NT4; Windows 2000; and Windows XP. Applies to ALL Linux Distributions.

David Allen

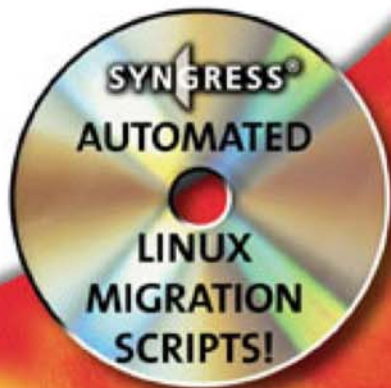
Andrew Scott

Herb Lewis

John Stile

Tim Tuck

Christian Lahti Technical Editor



Register for Free Membership to

s o l u t i o n s @ s y n g r e s s . c o m

Over the last few years, Syngress has published many best-selling and critically acclaimed books, including Tom Shinder's *Configuring ISA Server 2000*, Brian Caswell and Jay Beale's *Snort 2.0 Intrusion Detection*, and Angela Orebaugh and Gilbert Ramirez's *Ethereal Packet Sniffing*. One of the reasons for the success of these books has been our unique **solutions@syngress.com** program. Through this site, we've been able to provide readers a real time extension to the printed book.

As a registered owner of this book, you will qualify for free access to our members-only solutions@syngress.com program. Once you have registered, you will enjoy several benefits, including:

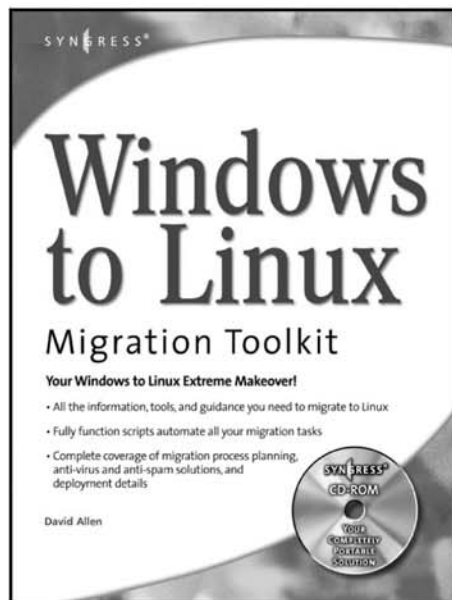
- Four downloadable e-booklets on topics related to the book. Each booklet is approximately 20-30 pages in Adobe PDF format. They have been selected by our editors from other best-selling Syngress books as providing topic coverage that is directly related to the coverage in this book.
- A comprehensive FAQ page that consolidates all of the key points of this book into an easy to search web page, providing you with the concise, easy to access data you need to perform your job.
- A "From the Author" Forum that allows the authors of this book to post timely updates links to related sites, or additional topic coverage that may have been requested by readers.

Just visit us at **www.syngress.com/solutions** and follow the simple registration process. You will need to have this book with you when you register.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there is anything else we can do to make your job easier.

Computer Resources Consulting, Inc.

<http://www.crconsulting.com> (800) 884-9885



The authors of this breakthrough book are available for consulting, support and training in migrating, networking, security, and systems. With over 25,000 successful migrations on five continents, Computer Resources Consulting has the expertise to ensure a smooth migration to systems that deliver outstanding value and increased productivity to any company. CR Consulting's satisfied clients include JPMorgan Chase, Charles Schwab, Credit Suisse First Boston, Applied Materials, NASA, and dozens of small businesses.

RELIABILITY * SCALABILITY * 24 x 7 AVAILABILITY

E-Mail
On the Web
By Telephone

MigrationHelp@crconsulting.com
<http://www.crconsulting.com>
(800) 884-9985



**CR Consulting
Incorporated***

Windows to Linux

Migration Toolkit

David Allen

Andrew Scott

Herb Lewis

John Stile

Tim Tuck

Christian Lahti Technical Editor

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively “Makers”) of this book (“the Work”) do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, “Career Advancement Through Skill Enhancement®,” “Ask the Author UPDATE®,” and “Hack Proofing®,” are registered trademarks of Syngress Publishing, Inc. “Syngress: The Definition of a Serious Security Library”™, “Mission Critical™,” and “The Only Way to Stop a Hacker is to Think Like One™” are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY SERIAL NUMBER

001	HJIR78N764
002	PO987JNHFG
003	82NJH24562
004	CVPLQ6WQ23
005	JKNN6653FL
006	VB5GHFF42
007	HJJEVBNK98
008	29WMKGHGG8
009	629TGHVB56
010	IMTVCX32X4

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Windows to Linux Migration Toolkit

Copyright © 2004 by Syngress Publishing, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in the United States of America
1 2 3 4 5 6 7 8 9 0
ISBN: 1-931836-39-6

Publisher: Andrew Williams
Acquisitions Editor: Jaime Quigley
Technical Editor: Christian Lahti and David Allen
Cover Designer: Michael Kavish

Page Layout and Art: Patricia Lupien
Copy Editor: Amy Thomson
Indexer: Rich Carlson

Distributed by O'Reilly Media, Inc. in the United States and Canada.
For information on rights and translations, contact Matt Pedersen, Director of Sales and Rights, at Syngress Publishing; email matt@syngress.com or fax to 781-681-3585.



Acknowledgments

Syngress would like to acknowledge the following people for their kindness and support in making this book possible.

Syngress books are now distributed in the United States and Canada by O'Reilly Media, Inc. The enthusiasm and work ethic at O'Reilly is incredible and we would like to thank everyone there for their time and efforts to bring Syngress books to market: Tim O'Reilly, Laura Baldwin, Mark Brokering, Mike Leonard, Donna Selenko, Bonnie Sheehan, Cindy Davis, Grant Kikkert, Opol Matsutaro, Steve Hazelwood, Mark Wilson, Rick Brown, Leslie Becker, Jill Lothrop, Tim Hinton, Kyle Hart, Sara Winge, C. J. Rayhill, Peter Pardo, Leslie Crandell, Valerie Dow, Regina Aggio, Pascal Honscher, Preston Paull, Susan Thompson, Bruce Stewart, Laura Schmier, Sue Willing, Mark Jacobsen, Betsy Waliszewski, Dawn Mann, Kathryn Barrett, John Chodacki, and Rob Bullington.

The incredibly hard working team at Elsevier Science, including Jonathan Bunkell, Ian Seager, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, Emma Wyatt, Rosie Moss, Chris Hossack, Mark Hunt, and Krista Leppiko, for making certain that our vision remains worldwide in scope.

David Buckland, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Pang Ai Hua, and Joseph Chan of STP Distributors for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

David Scott, Tricia Wilden, Marilla Burgess, Annette Scott, Andrew Swaffer, Stephen O'Donoghue, Bec Lowe, and Mark Langley of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji Tonga, Solomon Islands, and the Cook Islands.

Winston Lim of Global Publishing for his help and support with distribution of Syngress books in the Philippines.



Lead Author

David Allen (President, CRCI) is the lead author and technical editor of *Windows to Linux Migration Toolkit*. David started programming computers when he was eight years old, and has over two decades of experience in the computer field. David has been responsible for over 25,000 migrations on five continents. He has worked for international banks (Schwab, Credit Suisse, JPMorgan), technology companies, and government organizations including NASA. David has been an Open Source advocate for many years, and is a speaker at LinuxWorld and O'Reilly's Open Source Convention.

David founded Computer Resources Consulting, Inc. to provide consulting services to clients around the world. CRCI provides migration, support, training, and security consulting using open source solutions. For more information about CRCI and migration services provided by the authors of this book, navigate to www.crconsulting.com or phone (800) 884-9885.



Contributing Authors

Herbert Lewis has been a member of the Samba team since 1997. He currently works at Panasas Inc. where he helps to support the CIFS gateway portion of the product using Samba software. He previously worked at SGI developing and maintaining the Samba software, as well as several other open-source software products on the IRIX operating system. He holds a bachelor's degree from the U.S. Coast Guard Academy, as well as a master's degree and an engineer's degree from Stanford University.

John Streeton Stile is the Senior Technical Engineer for Pervasive Networks, designing Open Source solutions for Windows and Unix environments. With his bachelor of science in biochemistry from the University of California, Davis, and drug discover R&D experience, John applies the scientific method to research, engineering, and trouble shooting network and computer solutions.

John joined the computer industry in 1997, and began exploring Unix in 1999 after discovering that, unlike biology, there is always a solution to computer problems. He has worked with entities large and small, public and private, in various industries, including: Industrial Light and Magic, Adobe Systems, Ohlone College, Certicom, and Skyflow.

John would like to thank John H. Terpstra for offering advice, updating RPMs, and adding a personal touch during the writing of this book. John is the author of *Samba-3 By Example*, the soon-to-be-released *OpenLDAP By Example*, and is a contractor for Samba solutions.

James Stanger (PhD., CIW Master Administrator, Linux+, Security+, A+, CTP) is Vice President of Certification at ProsoftTraining. He is chair of the LPI Advisory Council, leads CIW exam development, and has helped developed certifications for Symantec and CompTIA. A prolific author, James has created titles for ComputerPREP, Symantec, Syngress, Sybex, and McGraw Hill. His titles include *Hack Proofing Linux*, the *E-mail Virus Protection Handbook*, and *Advanced Internet Services Management*. He is also an accomplished network consultant, where he specializes in security auditing, Windows to Linux migration, and LAMP-based e-commerce solutions.

Andrew Taylor Scott is a student of computer science and philosophy at City College of San Francisco and part-time Linux consultant for non-profits looking to leverage open source software within their organizations. Before he went back to school, he was working at Linuxcare Inc., a revolutionary organization providing

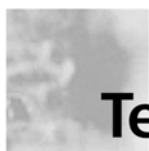
distribution-neutral technical support and professional services to enterprise-level companies seeking Linux solutions. While at Linuxcare, Andrew held the positions of Technical Support Engineer, Professional Services Consultant. He also served as Technical Writer, developing courseware in SGML for training Linux engineers in email systems and Web systems hosted on GNU/Linux. Taking full advantage of his time at Linuxcare, he learned as much as he could about GNU/Linux and the Free Software movement by involving himself in the development and release of several versions of the Linuxcare Bootable Business Card and the LNX-BBC mini-Linux distributions. He has since served as a consultant on several custom Linux solutions developing Web applications with Linux, Apache, MySQL, and PHP (LAMP).

Some of Andrew's clients include Thrasher Magazine, Theme-Co-op Promotions®, Fast Country, Institute for Collaborative Change, and Associated Students Councils at CCSF. He has completed several certifications in Linux system and network administration from Linuxcare University, including LNX-102, LNX-201, LNX-202, and LNX-301. He has also received Sun's Solaris System Administration I certification and has excelled in multiple in-depth classes regarding Unix and Linux operating system internals and C++ programming. Andrew has real-world experience deploying Linux servers for in-house networks, as well as collocated networks, establishing hardware and network needs, and designing appropriate Linux fixtures—often at a greatly reduced cost compared to commercial solutions. He is currently working on developing open source groupware with LAMP for virtual hosts.

Timothy Tuck is the President of Pervasive Networks and the Founder of the Hayward Linux Users group known as LinuxDojo.net. He specializes in Linux and Windows to Linux migrations. His company currently provides IT services for over 70 Companies in and around the Silicon Valley. Timothy's background includes positions in research and development as a Prototyper for Logitech, and as a Senior Engineering Tech and Lab Administrator at Cisco Systems. He has been working with computers for the last

20 years, using Linux on the desktop for the last 6 years. Timothy currently resides in Hayward, CA and is married to the most wonderful woman in the world, Louise Cheng.

Timothy would like to give special mentions and thanks to David Allen for coming up with the idea for this book, Jaime Quigley for her help during the writing process, Andrew Scott for his help during the editing of chapters, and to Rick Moen for the recommendation. A huge 'thank you' goes out to all of the hackers and programmers who contribute to Open Source Software. Their contributions have made all of this possible. To Linus Torvalds for giving the world the ultimate gift, Linux; the gift that keeps on giving. To the penguinheads from LinuxDojo.net, whose contributions to the users group keep it fun and real month after month. Special mention to Tim's lovely wife, who has given up everything to come half way across the world to put up with his endless hours of hacking around.



Technical Editor

Christian Lahti is a senior consultant with CRCI and has over 15 years experience in the IT industry. He is an expert in security, systems, and networking, having developed and implemented global IT infrastructures with a focus on Linux and open source, as well as providing consulting expertise for successful cross-platform integrations and interoperability. In addition, he is also skilled in database design as well as web development. Christian is a speaker and tutorial presenter at both LinuxWorld and O'Reilly's OSCON.

Contents

About the CDxxv
Forewordxxix
Chapter 1 Network Services Migration Roadmap1
Introduction2
Assessing the Current Infrastructure2
A Tale of Two Companies3
Inventorying the Servers5
Creating an Infrastructure Diagram6
Documenting Additional Assessment Information8
Determining Requirements for the Linux Infrastructure8
Creating a Functional Requirements Document8
Identifying Constraints10
Designing the Linux Infrastructure11
Creating a Post-Migration Infrastructure Design11
Testing the Linux Infrastructure13
Creating a Test Plan14
Deploying the Linux Infrastructure16
Migrating to the Linux Infrastructure16
Summary18
Solutions Fast Track18
Chapter 2 Core TCP/IP Networking Services21
Introduction22
Understanding IP Address Assignment Services23
Understanding DHCP Clients24
Obtaining the Initial DHCP Lease24
Renewing a DHCP Lease26
Releasing a DHCP Lease27

Configuring DHCP Servers	28
Understanding Name Resolution Services	31
Understanding File-Based Name Resolution	31
Understanding DNS Name Resolution	32
Understanding Dynamic DNS (DDNS)	33
Configuring BIND and DHCP for Dynamic DNS ...	33
Understanding Time Synchronization Services	34
Understanding Network Time Protocol	34
Understanding Linux NTP Clients	35
Understanding Windows Time Service Clients	36
Migrating MS-DNS/DHCP to Linux BIND/DHCPD ...	37
Migrating the DHCP Scopes and Options	38
Determining the DHCP Scopes and Options on Windows NT	38
Determining the DHCP Scopes and Options on Windows 2000	39
Recording the DHCP Scopes and Options	40
Configuring the ISC DHCPD Server	41
Migrating from Microsoft DHCP Services to Linux DHCP Services	42
Backing Out of a Failed DHCP Migration	42
Migrating the DNS Information	42
Installing and Configuring BIND	42
Transferring DNS Information	43
Migrating from Microsoft DNS Services to Linux DNS Services	44
Backing Out of a Failed DNS Migration	45
Summary	46
Solutions Fast Track	46
Frequently Asked Questions	49
Chapter 3 Directory Services	51
Introduction	52
Understanding LDAP and Directories	52
Understanding LDAP Terms	53
Understanding Directory Structure	53
Understanding Directory Objects	54

Connecting to a Directory Server	55
Understanding LDAP Queries	56
Understanding Microsoft Directory Services	57
Understanding Windows NT SAM	57
Understanding Exchange 5.5 Directory Services	58
Understanding Active Directory	58
Understanding OpenLDAP	59
Understanding OpenLDAP Server Daemons	60
Understanding OpenLDAP Utilities	60
Designing Linux-Based Directory Services	61
Designing a Directory Information Tree	61
Designing the OpenLDAP Infrastructure	62
Configuring and Testing the OpenLDAP Server(s)	64
Summary	67
Solutions Fast Track	67
Frequently Asked Questions	70
Chapter 4 Authentication Services	71
Introduction	72
Understanding Windows Authentication	73
Understanding Windows 98/NT Logon	73
Understanding Windows 2000/XP Logon	74
Understanding Linux Authentication	74
Understanding LDAP Authentication	75
Understanding /etc/passwd/shadow Authentication	76
Understanding NIS Authentication	77
Understanding Linux Client Authentication Settings	77
Understanding Name Service Switch (nsswitch.conf)	77
Understanding LDAP Configuration Settings (ldap.conf)	78
Understanding PAM	79
Designing Linux-Based Authentication Services	81
Designing Cross-Platform Authentication Services	82
Installing and Configuring Samba	84
Migrating from NT/Exchange or Active Directory	88
Preparing for Migration	88
NT / Exchange 5.5 Migration Path	89

Active Directory Migration Path	93
Migrating Windows Logon Authentication Files	97
Testing Authentication Services	97
Enabling Encryption	97
Summary	100
Solutions Fast Track	100
Frequently Asked Questions	103

Chapter 5 File Services105

Introduction	106
Understanding Windows File Systems	106
Understanding Windows File Allocation Table	
File Systems	106
Understanding Windows NTFS File Systems	109
Understanding Linux File Systems	112
Understanding Ext2/3	113
Resources	114
Understanding ReiserFS	115
Resources	116
Understanding Permissions Management (Access Control)	118
Understanding File Backup, Restore, and Replication	
Options	123
Identify Critical Files to be Backed Up	124
Decide on a Backup Method	125
Determine What Type of Backup Media to Use . . .	125
Backup Schedules	126
Plan to Rotate and Archive Media	127
AMANDA	129
AMANDA Backup Process	130
Installing AMANDA	132
On the Server	134
On Linux Clients	137
For Windows Clients	138
Set Up a Backup on the Server	138
Designing Linux-based File Services	139
Hardware Resources	139
Migrating File Services to Linux	140

Migrating the File Permissions	142
Support Tools Utility showwaccs	144
Solutions Fast Track	145
Understanding Linux File Systems	145
Understanding Permissions Management (Access Control)	145
Understanding File Backup, Restore, and Replication Options	146
AMANDA	146
Frequently Asked Questions	147
Chapter 6 Print Services	149
Introduction	150
Understanding Windows Print Services	150
Understanding Linux Print Services	151
Configuring Linux Printing Using BSD or SYSV	153
Configuring Linux Printing Using CUPS	155
Sharing Samba Printers	161
Understanding Automatic Printer Driver Downloading	170
Migrating Windows Print Services to CUPS/Samba	175
Summary	176
Solutions Fast Track	176
Frequently Asked Questions	178
Chapter 7 Messaging Services	181
Introduction	182
Understanding Microsoft Messaging Services	183
Understanding Exchange 5.5 Messaging	184
Understanding Exchange 2000 Messaging	186
Understanding Linux-Based Messaging Services	187
Sending and Receiving Internet E-mail	187
Understanding Linux Messaging System Components	189
Mail User Agent	190
Mail Access Agent (MAA)	192
Mail Transfer Agent (MTA)	193
Mail Delivery Agent	195
Mailstore	195

Exploring Open Source Messaging Server Software . . .	197
Sendmail	197
Qmail	198
Postfix	199
Exim	200
Courier Suite	200
Designing Linux-Based Messaging Services	205
Determining the Internet E-mail Architecture	205
Choosing Messaging Server Software	208
Determining How to Store the E-mail	210
Creating a Linux E-mail Server Diagram	211
Integrating Anti-Spam and Anti-Virus Services	212
Understanding Spam	213
Determining if an E-Mail is Spam	213
Whitelisting and Blacklisting	216
Understanding Viruses	216
Exploring Open Source Anti-Spam / Anti-Virus	
Applications	219
SpamAssassin	220
Clam AntiVirus	220
MailScanner	221
Designing Anti-Spam / Anti-Virus Services	221
Shared E-Mail / Anti-Spam / Anti-Virus	
Server Option	222
Dedicated Anti-Spam / Anti-Virus Gateway Option	222
Commercial Appliance Option	224
Outsourced Option	224
Migrating Information from Exchange to Linux	225
Preparing Exchange for Migration	225
Installing the Courier IMAP server suite	227
Migrating E-mail to Linux	228
Summary	230
Solutions Fast Track	231
Frequently Asked Questions	234

Chapter 8 Groupware and Calendaring Services . . .	235
Introduction	236
Understanding Exchange and Outlook Groupware and Calendaring Features	236
Understanding Linux-Based Groupware and Calendaring Services	238
Understanding eGroupware	238
Understanding Horde	239
Reviewing OPEN-XCHANGE (OX)	240
Understanding TWiki Collaboration	241
Migrating to Linux Groupware / Calendaring Services . .	242
Understanding Outport	243
Understanding Other Migration Methodologies . . .	244
Summary	245
Solutions Fast Track	245
Frequently Asked Questions	247
Chapter 9 Web Services: IIS vs Apache	249
Introduction	250
Background: HyperText Transfer Protocol	251
Understanding Microsoft's Internet Information Server . .	252
Getting Started	252
Default Index Page	253
Default Web Site	255
Virtual Directories	256
Security	257
SSL/TLS	258
Virtual Servers	260
Understanding Apache Web Server	262
Background of Apache	263
Installation	264
Starting, Stopping, and Checking Status	264
Configuration	265
Basic Configuration Options for the Main Server .	267
Default Web Page	270
Aliases and ScriptAliases	271
Security and Permissions	272

.htaccess Files	275
Virtual Hosting	276
Modules	282
Graphical Tools	284
Migrating Static Sites from IIS to Apache	286
Summary	287
Solutions Fast Track	287
Frequently Asked Questions	289

Chapter 10 Desktop Migration Roadmap291

Introduction	292
Assessing the Current Desktop Environment	294
User Types	294
Kiosk	294
Basic Knowledge Worker	294
Transactional Worker	295
Technical Worker	295
Advanced Knowledge Worker	295
Creating the Desktop Asset List	295
Cataloging File Formats	299
Functional Requirement Specification	300
Designing the Linux Desktop	301
Functional Replacement Specification	303
Building the Training Computers	304
Testing the Linux Desktop	305
Migrating Application Data and Profiles	305
Backing Up Desktop Systems	306
Installing Linux	307
Importing User Profiles and Preferences	307
Training the Desktop Users	307
Training Guidelines	308
Linux Desktop Differences	309
Filesystem Differences	310
Other Differences	310
Alternative Application Equivalents	311
Getting Help	312
Deploying the Linux Desktops	312

Documentation	313
Summary	314
Solutions Fast Track	314
Frequently Asked Questions	316
Chapter 11 Inside the Typical Linux Desktop	317
Introduction	318
Common Desktop Environments	318
Gnome	319
KDE	321
Common Features	323
Install Both, Make One the Default	323
Alternative Window Managers	323
The X Window System and Window Managers	324
X Window Servers versus Window Managers	325
Window Managers as Alternative Desktop Environments	327
E-Mail and Personal Information Management Clients ..	329
Evolution	330
Evolution, Microsoft Exchange, Novell GroupWise, and OpenExchange	332
KDE Suite/KMail	332
Kontakt	333
Aethera	333
Mozilla Mail/Thunderbird	334
Thunderbird	335
Sylpheed	335
Essential Information	336
Recommending E-Mail and PIM Software	336
Migrating Mail	337
Migrating from Outlook or Outlook Express	337
Importing Outlook Mail into Mozilla	338
LibPST	339
Importing Outlook Mail into Evolution	339
Document Standards	342
The Hard Way	342
Web Browsers	342

Mozilla	.343
Mozilla and Microsoft CHAP	.344
Firefox	.344
Galeon	.345
Konqueror	.346
Opera	.346
Migrating Bookmarks	.347
Browser Plug-Ins	.347
Macromedia Flash and Shockwave/Director	.348
RealPlayer	.348
Adobe Acrobat Reader	.349
Office Application Suites	.350
OpenOffice.org	.351
Limitations: Macros and PDF Files	.353
Future Plans	.354
StarOffice	.354
KOffice	.355
Hancom Office	.355
Running Windows Applications on Linux	.356
Compatibility Layer Software	.356
Wine	.357
Code Weavers' CrossOver Office	.358
Summary	.359
Solutions Fast Track	.359
Frequently Asked Questions	.361

Appendix A Introducing Network Analysis and Ethereal	.365
Introduction	.366
What is Network Analysis and Sniffing?	.366
Who Uses Network Analysis?	.369
How are Intruders Using Sniffers?	.370
What does Sniffed Data Look Like?	.372
Common Network Analyzers	.373
How Does It Work?	.378
Explaining Ethernet	.378
Understanding the OSI model	.380

CSMA/CD	.384
Hardware: Taps, Hubs, and Switches, Oh My!	.385
Port Mirroring	.388
Defeating Switches	.389
Detecting Sniffers	.391
Protecting Against Sniffers	.395
Network Analysis and Policy	.397
Summary	.398
Solutions Fast Track	.399
Frequently Asked Questions	.401

Appendix B Introducing Intrusion Detection

Systems and Snort	.403
Introducing Intrusion Detection Systems	.404
What Is an Intrusion?	.404
Legal Definitions	.405
Scanning vs. Compromise	.407
Viruses and Worms—SQL Slammer	.408
Live Attacks—Sendmail Buffer Overflow	.411
How an IDS Works	.411
What the IDS Is Watching	.411
How the IDS Watches Your Network	.422
How the IDS Takes the Data It Gathers and Finds	
Intrusion Attempts	.424
What the IDS Does When It Finds an Attack	
Attempt	.427
Answering Common IDS Questions	.429
Why Are Intrusion Detection Systems Important?	.430
Why Doesn't My Firewall Serve as an IDS?	.430
Why Are Attackers Interested in Me?	.430
Automated Scanning/Attacking Doesn't Care	
Who You Are	.431
Desirable Resources Make You a Target	.431
Political or Emotional Motivations	.432
Where Does an IDS Fit with the Rest of My	
Security Plan?	.433
Where Should I Be Looking for Intrusions?	.433

Operating System Security—Backdoors and Trojans	434
Physical Security	434
Application Security and Data Integrity	436
Correlation of All These Sources	437
What Will an IDS Do for Me?	437
Continuously Watch Packets on Your Network and Understand Them	437
Read Hundreds of Megs of Logs Daily and Look for Specific Issues	438
Create Tremendous Amounts of Data No Matter How Well You Tune It	438
Create So Much Data that If You Don't Tune It, You Might as Well Not Have It	439
Find Subtle Trends in Large Amounts of Data that Might Not Otherwise Be Noticed	439
Supplement Your Other Protection Mechanisms	439
Act as a Force Multiplier Competent System/ Network Administrator	440
Let You Know When It Looks Like You Are Under Attack	440
What Won't an IDS Do for Me?	441
Replace the Need for Someone Who Is Knowledgeable about Security	441
Catch Every Attack that Occurs	441
Prevent Attacks from Occurring	442
Prevent Attacks from Succeeding Automatically (in Most Cases)	443
Replace Your Other Protection Mechanisms	444
What Else Can Be Done with Intrusion Detection?	444
Fitting Snort into Your Security Architecture	444
Viruses, Worms, and Snort	445
Known Exploit Tools and Snort	445
Writing Your Own Signatures with Snort	446
Using an IDS to Monitor Your Company Policy	446
Analyzing Your IDS Design and Investment	446
False Positives versus False Negatives	447

Fooling an IDS	.447
IDS Evasion Techniques	.447
Return on Investment—Is It Worth It?	.449
Defining IDS Terminology	.450
Intrusion Prevention Systems (HIPS and NIPS)	.450
Gateway IDS	.450
Network Node IDS	.450
Protocol Analysis	.451
Target-Based IDS	.451
Summary	.452
Solutions Fast Track	.452
Frequently Asked Questions	.454

Appendix C Introducing Vulnerability Assessments and Nessus .455

Introduction	.456
What Is a Vulnerability Assessment?	.456
Why a Vulnerability Assessment?	.458
Assessment Types	.459
Host Assessments	.460
Network Assessments	.461
Automated Assessments	.461
Stand-Alone vs. Subscription	.462
The Assessment Process	.463
Detecting Live Systems	.463
Identifying Live Systems	.464
Enumerating Services	.464
Identifying Services	.466
Identifying Applications	.466
Identifying Vulnerabilities	.467
Reporting Vulnerabilities	.468
Two Approaches	.469
Administrative Approach	.469
The Outsider Approach	.470
The Hybrid Approach	.471
Realistic Expectations	.473
The Limitations of Automation	.475

Summary	476
Solutions Fast Track	477
Frequently Asked Questions	478
Index	481

About the CD

The CD-ROM accompanying this book contains the Windows to Linux Migration Toolkit (W2LMT) scripts and configuration files, as well as a chapter-by-chapter summary of the configuration files used by the fictional companies Acme Widgets and Ballystyx Engineering. Navigate to **index.html** for an easy-to-use chapter-by-chapter guide to the features of the CD

The following table lists each directory, chapter affiliation, and a description of the files located in that directory. It is important to note that although the package directory listed below does contain the latest versions of the Open Source tools used in the book at the time of publication, most Open Source projects develop and evolve at a rapid pace; the scripts may be updated or outmoded by the time you read this. You may want to obtain the latest versions from the appropriate websites, including the W2L MT (Windows to Linux Migration Toolkit), found at www.syngress.com/solutions and <http://sourceforge.net/projects/w2lmt>.

Directory	Chapter	Contents
Chap01	Network Migration Roadmap	-
Chap02	Core TCP/IP Services	DHCP/DNS and migration script configuration files
Chap03	Directory Services	OpenLDAP configuration files
Chap04	Authentication Services	Samba and directory/auth migration script configuration files

Continued

Directory	Chapter	Contents
Chap05	File Services	-
Chap06	Print Services	-
Chap07	Messaging Services	Mailbox migration script configuration files
Chap08 (Outport)	Groupware and Calendaring	Outlook Export tool
Chap09	Web Services	-
Chap10	Desktop Migration Roadmap	-
Chap11	Inside the Linux Desktop	-
Etc	ALL	Generic configuration files for w2lmt with comments
Package	ALL	Packaged binaries and compressed source files of many of the tools mentioned in the book, including w2lmt, and all the dependencies required to install them
Src	ALL	W2lmt scripts in source form
et_sn_ns	Appendix A	Ethereal
et_sn_ns	Appendix B	Snort
et_sn_ns	Appendix C	Nessus



Author Acknowledgments

While the author is the most visible source of work in the production of a book, there are many people that are equally important to the successful delivery of the book.

I would like to thank the fine people at Syngress publishing, particularly Andrew Williams and Jaime Quigley. Their knowledge of publishing and editing, along with their hard work and devotion, allowed me transform *Windows to Linux Migration Toolkit* from an idea into a book.

I would particularly like to thank Christian Lahti, the author of the *Windows to Linux Migration Toolkit* scripts, and packager of the accompanying CD. His late nights perfecting the scripts, his willingness to host the Iceberg lab (where Acme Widgets and Ballystyx Engineering meet the real world!), and his invaluable insight and support during the writing of this book contributed significantly to the quality of the material, making the toolkit a reality. People all over the world will be using his code to automate their migration from Windows to Linux. Thanks, Chris.

I would also like to thank all of my friends and family, particularly my parents. I extend my gratitude to Stephen Hoffman, Drayton Bowles, and Bob Cooley, for all of their support and understanding during the lengthy writing process.

Herb Lewis, Andrew Scott, James Stanger, Peter Thoeny, Sam Varshavchik, John Stile, and Tim Tuck were instrumental in writing a number of the chapters. Thanks, guys.

Thanks to Jeremy Allison for his support during the book writing process, and even more importantly, thanks for writing Samba!

—David Allen
October 7, 2004

Foreword

If the computer technology industry is the driving force behind the world's innovative global economy, then a computer's operating system is the engine that propels the vehicle forward. Since 1981, Microsoft has been the primary builder of that engine in the consumer and corporate marketplace.

Microsoft's operating system and application software have addressed the needs of the market for over twenty years, becoming one of America's great success stories. Microsoft has thrived in lean and healthy economic times in a rapidly changing industry. But, just as Henry Ford's competitors chased the Model T's market hold and monopoly, a group of penguins is quickly gaining ground on the behemoth juggernaut from Redmond.

Linus Torvalds didn't set out to compete with Microsoft when he first posted his notice of his new kernel on Usenet in August 1991. But on that day, the IT world changed forever. From its humble beginning, Linux has risen to great prominence, with tens of millions of users worldwide. On August 25, 1991, Linus presented his ideas for free open source to his peers. He writes:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroup: comp.os.minix

Subject: What would you like to see most in minix?

Date: 25 Aug 91 20:57:08 GMT

Hello everybody out there using minix-

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like

any feedback on things people like/dislike in minix; as my OS resembles it somewhat (same physical layout of the file-system due to practical reasons)among other things.

I've currently ported bash (1.08) an gcc (1.40), and things seem to work. This implies that i'll get something practical within a few months, and I'd like to know what features most people want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Instead of trying to capitalize on his intellectual property, Linus decided to give the fruits of his labor to the world for free, and ensure that others could contribute to the project that would later become the world's most popular open source operating system. He did this by licensing Linux under an open source license, the GNU Public License (GPL).

There are many benefits to using open source software (OSS), noted by David Wheeler at www.dwheeler.com/oss_fs_why.html

1. OSS protects users from risks and disadvantages from a single supplier.
2. OSS protects users from licensing litigation and management costs.
3. OSS has greater flexibility than closed source (proprietary) software.
4. There are positive social, moral, or ethical implications to using OSS.
5. There is ample evidence that OSS encourages, not quashes, innovation.

David Allen, President and founder of CR Consulting, is another proponent of Linux and Open Source. By authoring *Windows to Linux Migration Toolkit*, David provides a path to migrate to the same or better functionality using open source instead of proprietary technology. Distilling information scattered among many sources, this book provides a way to get off of the upgrade treadmill with step-by-step instructions, best practices, and automated migration scripts to migrate from Windows to Linux. With over 25,000 migrations performed on five continents, David Allen's knowledge and expertise provides valuable real-world experience in the areas of project management, Linux-based systems design, and migration planning and implementation. Many of these techniques have been utilized at companies such as JPMorgan Chase, Applied Materials, and NASA.

As David Wheeler wrote, businesses and users do not want to be held hostage by vendors. Businesses often prefer to buy products that exist among a group of competing suppliers because competition reduces their risk—they can always switch to another supplier if they're not satisfied, if another supplier offers better prices, or if the original supplier goes out of business. This translates into an effect on the products themselves: if customers can easily choose and switch between competing products, the products' prices go down and their quality goes up, and the customer is benefited.

This book makes migrating from Windows to Linux easier than anyone thought possible. Go forth and migrate!

—*Drayton Bowles*
October 2004

Network Services Migration Roadmap

Solutions in this Chapter:

- Assessing the Current Infrastructure
 - Determining Requirements for the Linux Infrastructure
 - Designing the Linux Infrastructure
 - Testing the Linux Infrastructure
 - Deploying the Linux Infrastructure
 - Migrating to the Linux Infrastructure
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

This chapter will teach you about planning and managing a Windows to Linux migration of network services. If you are a systems administrator (sysadmin) with no interest in migration planning and management, you may safely skip this chapter, particularly if someone else is handling these concerns. If you are an IT manager, a project manager, or a migration consultant, this chapter will be of great interest to you. It provides a framework and roadmap for preparing successful Windows to Linux migration project and gives step-by-step instructions to help ensure a hassle-free migration.

Management of migration projects is a subset of general project management. Like most software deployment projects, migration projects include assessment, requirements, design, testing, and deployment phases. The information in the rest of this chapter addresses the project management concerns that exist for each project phase, and offers best practices methodology specific to Windows to Linux migration projects.

While migration planning and management activities can represent a significant up-front investment in a project, they inevitably pay off later in the form of a smooth, hassle-free migration. This level of planning and management is especially important for larger projects—a few extra hours of preparatory activity during an early phase of a project can often save a day or more of work during a later phase.

Assessing the Current Infrastructure

Migration projects are similar to a road trip. Both feature a starting point, a journey, and a destination. The first step in planning a migration is learning about the starting point. This is accomplished by performing a written assessment of the current IT environment of the organization. An assessment gives detail to your starting point and helps determine the scope of the migration project.

At a minimum, your assessment should include all data related to the areas of migration on which you will focus. Because an infrastructure assessment usually represents a significant one-time effort, it makes sense to gather as much useful data about your systems and network infrastructure as time permits.

A Tale of Two Companies

To better illustrate the different needs and demands your company's infrastructure will necessitate during a Windows to Linux changeover, we will follow two very different companies through a fictitious migration. Each chapter will track the migration progress of Acme Widgets and Ballystyx Engineering; identifying real-life scenarios, benefits, challenges, and estimated costs that an administrator can identify with.

Acme Widgets Profile

Full Name: Acme Widgets Manufacturing, Inc.

Location: Knoxville, Tennessee

Years in Business: 5

Number of Employees: 6

Migration Consultant: Sam Aster

For the past five years, the custom widget manufacturing business, Acme Widgets, has prospered. The whitebox Exchange server and Windows desktops have met their computing needs as they've grown from three to six employees.

A month ago, a computer became infected with a virus and wouldn't boot. Since the virus had wiped out the hard drive, Sam asked Acme Widgets for the installation media to reinstall the OS. A quick search turned up the fact that Acme Widgets did not own installation media or licenses for any of the software that they're running. A contracted IT professional had used pirated serial numbers to install thousands of dollars of software.

To get the desktop running, Windows 2000 Desktop was purchased so they could legally install Windows on the virus-wiped machine. Because the company couldn't afford a Microsoft Office license, Sam installed OpenOffice on the other machines.

After reading over the "Report Piracy" form on the Business Software Alliance website, Sam performed an assessment and concluded that migrating to Linux would save Acme Widgets thousands of dollars. Sam's plan is to migrate the unlicensed desktops to Linux running Mozilla, OpenOffice, and Evolution. He will build a new server to replace the Exchange Server.

Ballystyx Engineering Profile

Full Name: Ballystyx Global Semiconductor Engineering, Inc.

Locations: Silicon Valley, USA; Bangalore, India

Years in Business: 10

Number of Employees: 76

IT Manager: Vijay Srinivasan

For over ten years, Ballystyx Engineering has been building chips for trajectory and ballistic computations. Their business is stable and established.

Over the years, they have experienced a growing problem with e-mail spam and viruses. The problem has gotten so bad that some users spend twenty minutes or more each day reviewing and deleting spam. Because of the productivity cost, the executive staff asked IT Manager, Vijay, to evaluate adding e-mail anti-spam and anti-virus protection.

Since Ballystyx Engineering uses Exchange for messaging, Vijay contacted anti-spam and anti-virus software vendors with Exchange-integrated products. He found out that it would cost at least \$7000 to add these features to Exchange, so Vijay decided to prepare a proposal to add open source anti-virus and anti-spam software to Hydrogen, the company's Linux server. Although the software might slow the server somewhat, he could address the spam and virus problem without paying costly software licensing fees.

While reviewing software license purchases for his report to the executive staff, Vijay discovered that although the Exchange server and Outlook clients were properly licensed, no Exchange client access licenses (CALs) had been purchased. A quick check of the Exchange server listed licensing for 999 users, a number that couldn't possibly be right. Someone had fudged the licensing entry and failed to purchase Exchange CALs costing over \$5000.

In addition to the spam and virus problem, he now had a much bigger dilemma—fork over \$5000 or migrate away from Exchange. It looked like open source applications could provide the Exchange-like features that Ballystyx Engineering needed, without Exchange-like prices. Vijay knew that open source was capable, but he didn't know that it could now replace an entire Exchange and Active Directory infrastructure. Ballystyx Engineering could eliminate their Windows Server infrastructure, transfer all network applications and migrate completely to Linux.

Inventorizing the Servers

The first step in performing a migration assessment is creating an inventory of all servers (see Tables 1.1 and 1.2). The inventory must include at least the server name, IP address, network services applications, operating system revision, and hardware details.

Table 1.1 Acme Widgets Server Inventory (Small Company)

Name	OS	Software	Memory	CPU	IP Address
SERVER1	WinNT SP6a	PDC, File/Print Exchange,	384 MB	P3-866Mhz	192.168.100.2

Table 1.2 Ballystyx Engineering Server Inventory (Medium - Large Company)

Name	OS	Software	Memory	CPU	IP Address
HYDROGEN	Debian (unstable)	Apache, PostgreSQL (postgres)	2 GB	2 x 1.4Ghz	192.168.1.10

Continued

Table 1.2 Ballystyx Engineering Server Inventory (Medium - Large Company)

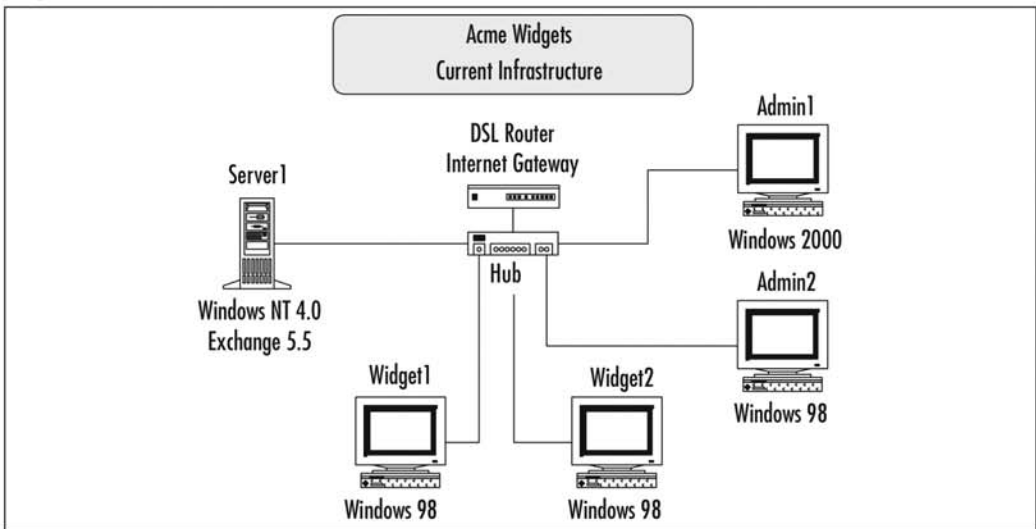
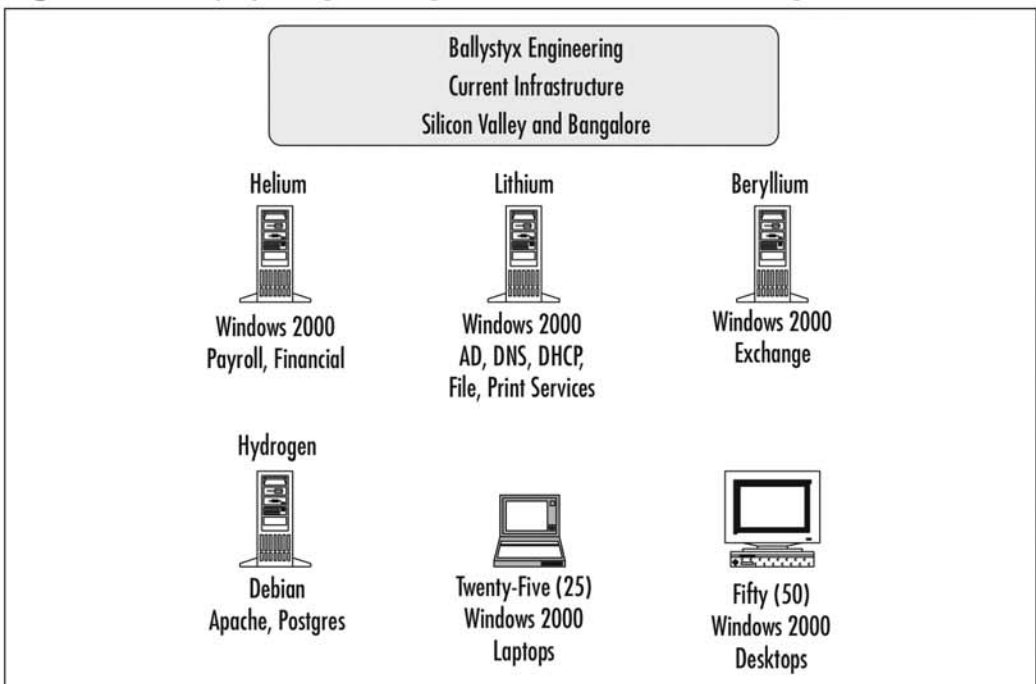
Name	OS	Software	Memory	CPU	IP Address
HELIUM	Win2K SP3	Payroll and Financial Processing	2 GB	1 x 1.4Ghz	192.168.1.11
LITHIUM	Win2K SP3	AD, DNS, DHCP, File/Print	1 GB	1 x 2Ghz	192.168.1.12
BERYLLIUM	Win2K SP3	Exchange	2 GB	1 x 2Ghz	192.168.1.13

The information gathered at this early phase will be used to plan and implement all other phases of the project, so it is critical to ensure the information is correct. When erroneous data becomes part of a project during the early planning phases, it tends to have a cascading effect as other erroneous data is generated from the original error.

In addition to creating a server inventory, it may be useful to create a desktop inventory, particularly if a migration to Linux desktops is planned. A desktop inventory can also help identify server requirements in many cases. See Chapter 11, “Desktop Migration Roadmap,” for information about creating a desktop inventory.

Creating an Infrastructure Diagram

A systems and network diagram (see Figures 1.1 and 1.2) of the current infrastructure provides a visual representation of the starting point for the migration. Visio and Dia (www.gnome.com/projects/dia) can be used for producing diagrams of this type.

Figure 1.1 Acme Widgets Current Infrastructure Diagram**Figure 1.2** Ballystyx Engineering Current Infrastructure Diagram

Documenting Additional Assessment Information

If time is available and thorough assessment documentation is desired, the following items should also be recorded:

- Hardware asset tag and/or serial number
- Hardware manufacturer and model
- Computer disk capacity and utilization
- Networking equipment manufacturer, model, and configuration

Determining Requirements for the Linux Infrastructure

With the assessment of the existing infrastructure complete, the next step is to determine requirements for the new Linux infrastructure. In all cases, we will try to provide at least similar functionality to the Windows infrastructure that we are replacing. In many cases, Linux will offer enhanced capabilities that are easy to leverage, particularly when the requirements are identified early in the project.

In general, requirements have to be determined based on business needs. It may be tempting to make a software feature a requirement simply because the software supports it. However, there are costs associated with the deployment and ongoing support of additional features, and most for-profit businesses are driven by these cost factors. When this type of situation is encountered, it is often useful to prepare a “must-have” list and a “wish” list, or to simply assign a priority to each requirement. If the costs (time, training, person-hours, support) of a requirement are minimal, it can be feasible to include it in the project even if it is a lower priority.

Creating a Functional Requirements Document

A functional requirements document, featured in Table 1.3, describes the functionality an infrastructure must provide in order to be considered acceptable. The list of functional requirements will later be used to create the test plan.

Table 1.3, shows the functional requirements document for Acme Widgets. The document is organized into network service groups, in the same way the

network service migration chapters of this book are organized. When you use this template to create your own functional requirements document, it will be easy to follow along with the network services migration methodology presented in this book.

Table 1.3 Acme Widgets Functional Requirements Document

Services	Requirements
1. Core Networking Services	
1.1 IP Address Assignment	Use static IP addresses (no DHCP)
1.2 Name Resolution	Use host file name resolution
1.3 Time Synchronization	Install time synchronization software on all computers
2. Directory Services	
2.1 Directory Services	Install and configure OpenLDAP
2.2 Directory Services Migration	Migrate information from Exchange and NT SAM to OpenLDAP
3. Authentication Services	
3.1 Authentication Services	Install and configure Samba authentication services
3.2 Authentication Services Migration	Users must change passwords
4. File Services	
4.1 File Services	Install and configure Samba file services
4.2 File Services Migration	Copy all file data to new server and set up network shares
5. Print Services	
5.1 Print Services	Install and configure CUPS.
5.2 Print Services Migration	Migrate print queue information from Windows
6. Messaging Services	
6.1 MTA Services	Install and configure Courier-MTA
6.2 Mailstore (MAA) Services	Install and configure Courier-IMAP
6.3 Webmail Services	No webmail requirements
6.4 Anti-Spam	Provided by ISP
6.5 Anti-Virus	Provided by ISP

Continued

www.syngress.com

Table 1.3 Acme Widgets Functional Requirements Document

Services	Requirements
6.6 Messaging Services Migration	Migrate Jim’s mailstore information
7. Groupware and Calendaring Services	
7.1 Calendaring Services	Install and configure calendaring server
7.2 Groupware Services	To Be Determined
7.3 Calendaring Migration	Migrate Jim’s calendar information
7.4 Groupware Migration	To Be Determined
8. Web Services	No Web Services Requirements

Identifying Constraints

In addition to business and/or technical requirements, there may be legal, cost, time, organizational, or other constraints. Constraints may necessitate the modification or elimination of certain requirements, or change the priority or scheduled sequence of events. Table 1.4 illustrates examples of constraints and the necessary steps it may take to troubleshoot unexpected issues.

Table 1.4 Migration Constraints and Adjustments

Constraint	Example	Solutions
Cost	There is not enough money for four new servers. The company can only afford one at this time.	Instead of using new servers for the Linux infrastructure, all but one of the existing Windows servers will be recycled into Linux servers. One Windows server will be used as a test lab computer. Three spare desktop workstations will now be required as test lab computers
Legal	Due to confidentiality issues, executive e-mail and non-executive e-mail must not be stored on the same physical server.	An additional server must be purchased

Continued

Table 1.4 Migration Constraints and Adjustments

Constraint	Example	Solutions
Time	Due to Tom's vacation, the migration cannot be performed between February 2 and February 16.	The migration will be rescheduled to the first weekend following February 16
Procedural	Due to corporate month-end billing procedures, no backups or reboots of billing systems can occur on the last day of the month.	Backups on billing servers will be suspended on the last day of the month. Any maintenance on billing servers requiring a reboot will not be scheduled on the last day of the month.

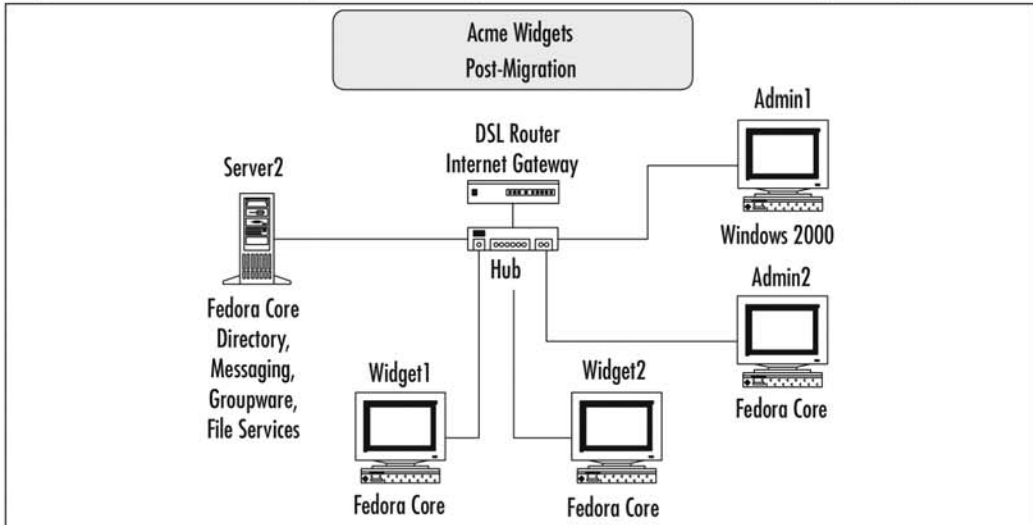
Designing the Linux Infrastructure

With the requirements established, you can begin the high-level design of the Linux infrastructure. This type of design work involves determining the overall server and network strategy for the post-migration infrastructure. You will decide which systems will be retained and which systems will be replaced, upgraded, or added.

At this point in the migration, it is too soon to determine the low-level details of the new infrastructure. As you progress through the chapters in this book, you will receive detailed design guidelines for each network service component.

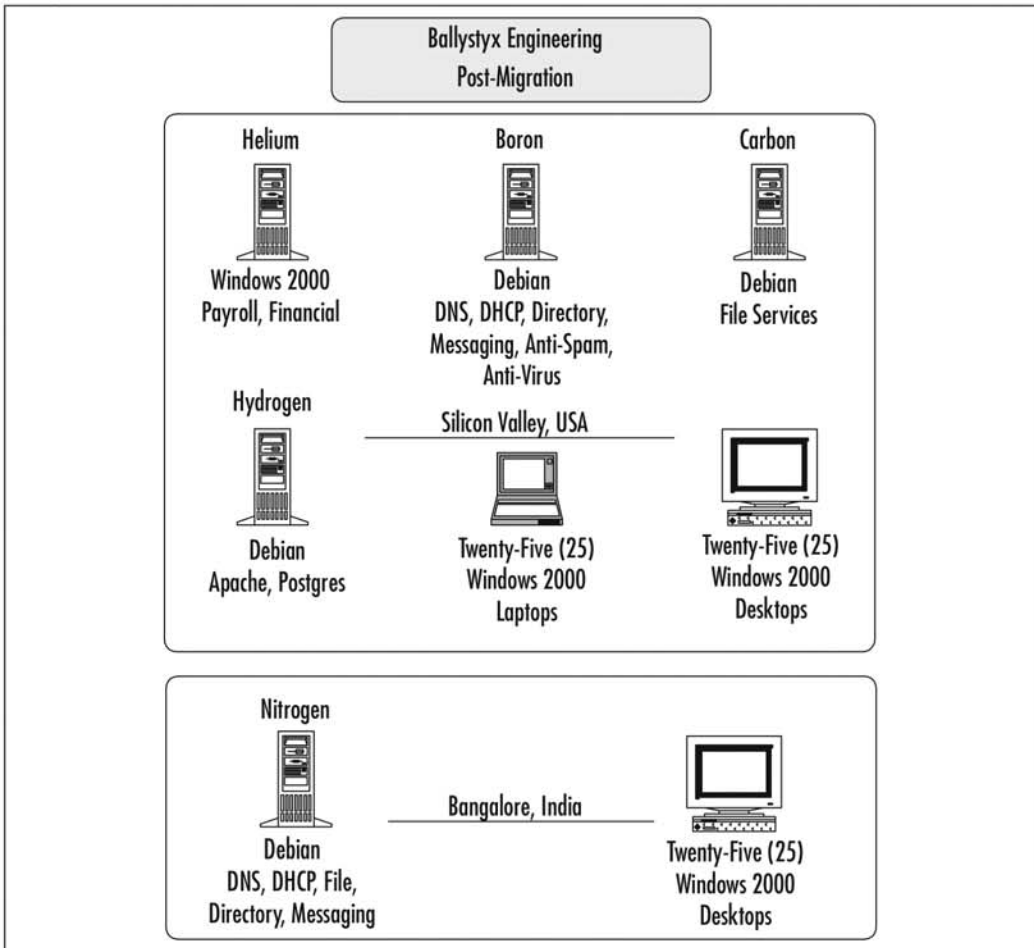
Creating a Post-Migration Infrastructure Design

Designing the new infrastructure will require determining the number of servers, placement of servers, and specific services running on each server. For small companies, one or two servers is often enough to cover the needs of the users. Figure 1.3 illustrates a post-migration infrastructure design suitable for a small company.

Figure 1.3 Acme Widgets Post-Migration Infrastructure Design Diagram

For larger companies with headquarters and branch offices, a typical arrangement is that the corporate database servers, intranet servers, messaging servers, and the like are housed at headquarters, and each branch office (of sufficient size/bandwidth/users) contains one server that runs file services and possibly authentication, directory, and/or e-mail services for the users at the branch office (See Figure 1.4). An arrangement of this type will ensure that a branch office can remain productive and mostly functional even during a WAN and/or Internet outage.

Figure 1.4 Ballystyx Engineering Post-Migration Infrastructure Design Diagram



Detailed design work involves the documentation of server and application configuration based on the requirements specifications. Each network services chapter in this book provides specific guidelines for network service application configurations in various types of environments.

Testing the Linux Infrastructure

Just as a commercial truck driver would test a new truck prior to using it on an important journey, it makes sense to thoroughly test the new Linux infrastructure prior to placing it into production. A company must be sure that the new

infrastructure will prove as capable as, or more capable than, the infrastructure it replaces, as well as ensuring the infrastructure meets or exceeds the functional requirements.

A test plan documents how each of the requirements will be tested. As each infrastructure component is deployed, it must be tested according to the procedures in the test plan. The infrastructure will be considered acceptable only after it has successfully passed all of the tests.

Creating a Test Plan

To create a test plan, leverage the functional requirements document you created earlier. Each feature in the functional requirements becomes an entry in the test plan. For complex features, the testing procedure may be lengthy.

A test plan (see Table 1.5) is a document that contains the requirements being tested and a description of how to test each item. Test plan detail varies with the complexity of the requirements and environment, the experience and skill level of the persons performing the work, and the meticulousness of the project manager.

At Acme Widgets, the procedure to test e-mail functionality is to send a test e-mail to the Internet and internal users. Since Sam is an experienced sysadmin, he knows to send a message to his Gmail e-mail address and verify that it was successfully delivered by opening and reading it. Because of the simplicity of the environment—and the fact that Sam is the sole person working on the migration project—the short description is sufficient.

In a larger company where the environment and service level requirements are more complex, the procedure to test e-mail functionality may be so lengthy as to require a separate document. A test procedure in Ballystyx Engineering’s test plan reads, “Ensure that a 4KB e-mail sent from a workstation in Silicon Valley reaches the e-mail server in Bangalore in less than 5 minutes.”

Table 1.5 Acme Widgets Test Plan

Services	Test Procedure
1. Core Networking Services	
1.1 IP Address Assignment	Use ping to test IP connectivity.
1.2 Name Resolution	Ping each host by name.
1.3 Time Synchronization	Verify the clock on each computer is correct.

Continued

Table 1.5 Acme Widgets Test Plan

Services	Test Procedure
2. Directory Services	
2.1 Directory Services	Use <code>gq</code> to verify LDAP queries are operational.
2.2 Directory Services Migration	Use <code>gq</code> to verify that entries in Exchange and NT SAM have been properly migrated to OpenLDAP.
3. Authentication Services	
3.1 Authentication Services	Verify that users can logon to Windows and Linux desktops.
3.2 Authentication Services Migration	Verify that users have changed their passwords from the default.
4. File Services	
4.1 File Services	Use <code>smbmount</code> to verify file services connectivity.
4.2 File Services Migration	Verify that all files copied successfully.
5. Print Services	
5.1 Print Services	Print a test page.
5.2 Print Services Migration	Verify that all computers can print properly.
6. Messaging Services	
6.1 MTA Services	Send a test e-mail to the Internet and internal users.
6.2 Mailstore (MAA) Services	Use Outlook and Evolution to verify mailstore access.
6.3 Webmail Services	No webmail requirements.
6.4 Anti-Spam	Provided by ISP.
6.5 Anti-Virus	Provided by ISP.
6.6 Messaging Services Migration	Verify that Jim's Exchange mailbox has been properly migrated.
7. Groupware and Calendaring Services	
7.1 Calendaring Services	Create a test appointment and meeting.
7.2 Groupware Services	To Be Determined.

Continued

www.syngress.com

Table 1.5 Acme Widgets Test Plan

Services	Test Procedure
7.3 Calendaring Migration	Verify that appointments and meetings have copied as expected.
7.4 Groupware Migration	To Be Determined.
8. Web Services	No Web Services Requirements

The use of a test lab is highly recommended. A test lab provides a networked environment that is separated or isolated from the production network. Changes to hardware and software configurations should be tested in the lab prior to being tested on the production network or deployed to production users.

Deploying the Linux Infrastructure

Following successful tests in the lab, the Linux infrastructure may be deployed on the production network for additional testing. When it seems certain that the new infrastructure is functioning properly, a sample group of users (called the *pilot group*) can test the features of the new infrastructure to determine if it works properly from their perspective. A best practice is to provide each pilot user with testing instructions and a feedback form, and then follow up with an in-person interview to review the feedback. Most pilot groups initially include IT personnel and technically savvy users, then expand to include mainstream users. It is important that the pilot group contain a representative sample of the overall group so that issues will be identified and solved before migration begins.

Migrating to the Linux Infrastructure

When you are certain that the new Linux infrastructure works as expected in the production environment, you can begin migrating users and systems to the new infrastructure. IT personnel and the pilot group are typically the first people who are migrated to the new infrastructure. There are a number of factors that can determine which users are migrated next:

- Need for new features offered by Linux (capabilities-driven)
- Business unit or department (business group-driven)
- Office location (geography-driven)

There are a number of ways to manage migration scheduling, and the importance (and complexity) of migration schedule management increases with the size and number of groups to be migrated. In larger migration projects, it is often best to appoint a stakeholder and/or point of contact whose job it is to manage communication between IT and the group or office s/he represents. This person can help keep track of who is ready to be migrated and the best dates and times to schedule migrations. The ideal arrangement is to have the business group points of contact be responsible for creating lists of users to be migrated so the IT group can concentrate on the actual migration process and supporting the new users. The migration process will most successful if business groups and IT are communicating well and the responsibilities of each group have been clearly defined and accepted.

Some types of migrations, such as authentication services, may be transparent to the end user, or at least transparent after a logout or reboot. While end users will have to be informed that changes are occurring, there may be no action required on their part. In other cases, migrations might be more disruptive, as in the case of migrating home directories to a new server with a new name or changing to a new e-mail application or web browser. These types of changes must be carefully scheduled and properly communicated. In cases where there are large numbers of users facing migrations, it makes sense to automate as much of the work as possible. Automating migration processes will help to minimize IT staff work requirements and reduce the possibility of human error.

Remember to plan for training. If new applications are being deployed or current processes are changing, it is important to communicate this knowledge and provide the affected people with updated documentation and training. The project will proceed more smoothly and people will feel better about the migration because they know their needs are being fully considered.

And finally, in all cases of migration planning, follow appropriate change management procedures. Require that all proposed changes be fully documented. Schedule regular meetings between decision makers and the technical staff to ensure proposed changes are discussed and understood prior to signoff.

Summary

Migration planning and management is an important part of a successful Windows to Linux migration. In larger projects, it is a critical part. Your initial investment in these planning activities will help ensure a smooth migration. While this may seem like a lot of work, the included templates will enable you to finish this job quickly and easily.

After reading this chapter you should have a good idea of the planning and management steps required in a Windows to Linux network services migration. If you have been following along with the instructions, you will also have a completed assessment diagram, server inventory, functional requirements document, high-level Linux infrastructure design, and test plan. You are on your way to achieving a successful Windows to Linux network services migration!

Solutions Fast Track

Assessing the Current Infrastructure

- ☑ Inventory the servers using the included templates.
- ☑ Create an infrastructure diagram of the current infrastructure.
- ☑ Document additional assessment information, if feasible.

Determining Requirements for the Linux Infrastructure

- ☑ Create a functional requirements document using the included templates.
- ☑ Identify constraints. Constraints often take the form of restrictions in project budgets or schedules.

Designing the Linux Infrastructure

- ☑ Create a post-migration infrastructure design diagram. The diagram should specify the number and placement of servers, as well as which network services are running on each server.

- ☑ Detailed design work will occur later in the migration process.

Testing the Linux Infrastructure

- ☑ Testing of the new infrastructure is required prior to deployment. Create a test plan to assist with the testing process.

Deploying the Linux Infrastructure

- ☑ Following successful testing in the lab, deploy the Linux infrastructure on the production network for additional testing.
- ☑ Allow a sample group of users (the pilot group) to test the Linux infrastructure to determine if it works from the end user perspective.

Migrating to the Linux Infrastructure

- ☑ When the pilot group has accepted the Linux infrastructure, you can begin to migrate other users and systems to the new infrastructure.
- ☑ Migration scheduling works best when the responsibilities of the business units and IT are clearly defined and accepted prior to commencing detailed migration planning or implementation.

Core TCP/IP Networking Services

Solutions in this Chapter:

- Understanding IP Address Assignment Services
 - Understanding Name Resolution Services
 - Understanding Time Synchronization Services
 - Migrating MS-DNS/DHCP to Linux BIND/DHCPD
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Internet Protocol (IP) addressing forms the basis of nearly all networks in use today, including the Internet. It is assumed that the reader understands (or employs someone who understands) the following fundamental Transmission Control Protocol (TCP)/IP knowledge:

- IP addresses and subnets
- IP packet broadcast and routing
- TCP/UDP sockets and syntax
- Network troubleshooting tools such as ping, traceroute, nmap, telnet, and tcpdump
- DNS servers, record types, and query tools such as nslookup and dig

If these networking concepts are not familiar, you may wish to read one of the Domain Name Service (DNS) and/or TCP/IP primers available on the Internet, or consider the purchase of a book about TCP/IP fundamentals to assist you with the comprehension of the following subject matter. O'Reilly's "DNS and BIND" provides a detailed guide to Linux-based DNS services.

This chapter addresses some of the basic network services found in nearly all businesses utilizing TCP/IP networks. DNS, DHCP (Dynamic Host Control Protocol), and time synchronization services are typically the first services running on a network because they form the core requirements needed to be able to utilize all other network services mentioned in this book. Nowadays, some of these core services may be provided by appliances or embedded devices, but this book will specifically examine Linux-based DHCP/DNS services. The discussion of time services includes the proper setup necessary to take advantage of the many publicly available Internet time servers.

The DHCP, DNS, and time services are briefly discussed below. The rest of the chapter will teach the basics of how these services work on Microsoft and Linux platforms, and explain how to migrate from Microsoft DNS/DHCP to Linux-based BIND/DHCPD services.

DHCP is the network protocol that dynamically assigns IP addresses and network configuration parameters to properly configure hosts. While most servers use static IP addresses, workstations generally receive dynamically assigned IP addresses, as well as other network information that allows them to communicate with other hosts on the network. Nearly all organizations use DHCP servers to

assign dynamic network information. We will examine the process of obtaining an initial DHCP lease, as well as the DHCP lease renewal process.

Name resolution services are another important part of computer network infrastructure. The ability to translate computer host names from easily-remembered English words into IP addresses forms a core usability underpinning of all sizes and types of computer networks, from a small business LAN to the global Internet. Without name resolution services, people would need to use difficult-to-remember IP addresses instead of easy-to-remember names. DNS provides these capabilities to Windows and Linux. We will examine the configuration of both Windows and Linux DNS services to facilitate migrating these services to your Linux platform.

The remaining essential core network service explored in this chapter is the ability to synchronize the clocks on all computers. While parts of a computer network may run properly with unsynchronized clocks, some network services require computer clocks to be synchronized within a few minutes of each other. In addition, it can be very difficult to analyze and correlate events when examining logfiles from computers with unsynchronized clocks. One of the top IT recommendations of the FBI is that all computer clocks be synchronized so data trails can be more easily followed during forensic analysis.

Understanding IP Address Assignment Services

On the Internet and in corporate environments, most servers use static IP address assignment. Static IP address configurations differ from dynamic addresses in these fundamental ways:

Table 2.1 Differences in Static and Dynamic IP Configurations

Static IP Address Assignment	Dynamic IP Addresses Assignment
IP address information rarely changes	IP address information expected to change
IP address changes following move or maintenance	Change in physical location changes IP address information
IP address information stored locally	IP address information cached locally, stored externally
Does not require external server	IP address information assigned by DHCP server

Most laptops and desktops do not utilize a static IP address. Instead, the IP address is leased from a DHCP server. Following a reboot or the expiration of a lease, the workstation will contact the DHCP server to renew the lease or obtain a new IP address. The main benefit of this setup is that the network configurations do not have to be set up individually for each client, and any network changes that occur in the future can be propagated to the clients when they renew their network lease, as discussed below.

Understanding DHCP Clients

A DHCP client is any network device – computer, printer, or network-enabled guitar – that makes use of DHCP to obtain its IP address information. When a DHCP client's network interface is started, the device will attempt to lease an IP address. The procedure to lease a new IP address is a four-step process, and is often referred to as the four-way handshake.

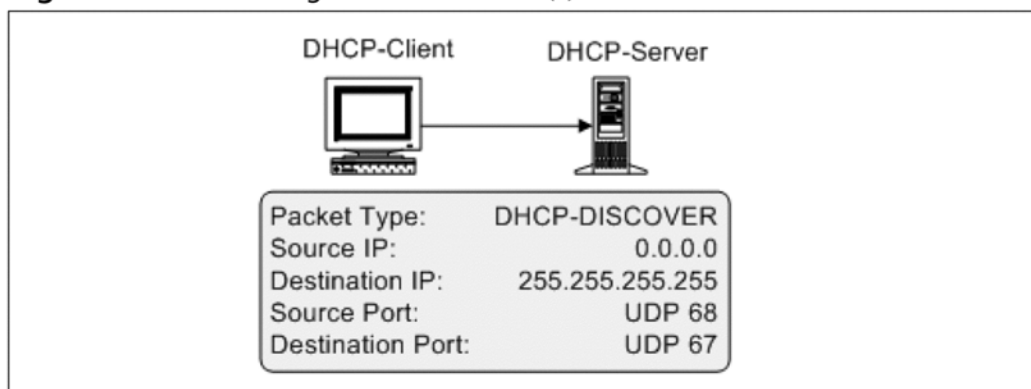
Obtaining the Initial DHCP Lease

The first time a computer is brought online, no information is known about DHCP servers. Therefore, the computer must send a broadcast request to discover a DHCP server.

Discovering the DHCP Server(s)

To initiate the DHCP lease process, the client sends a DHCP-DISCOVER packet in broadcast mode from UDP (User Datagram Protocol) port 68 to UDP port 67. This packet has a source address of 0.0.0.0 and a destination address of 255.255.255.255.

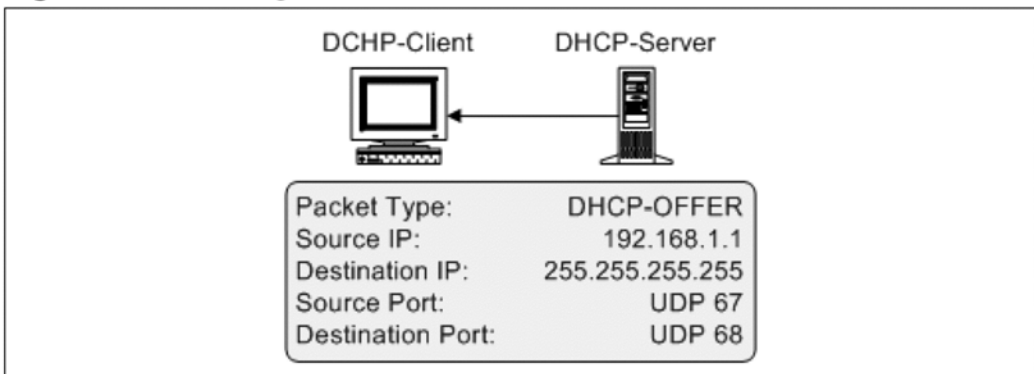
Figure 2.1 Discovering the DHCP Server(s)



Receiving the DHCP Lease Offer(s)

A DHCP server listening on the local subnet responds to the DHCP-DISCOVER broadcast request. On larger networks, a switch, router, or another DHCP relay agent may forward DHCP packets. If the request is valid and IP addresses are available, a DHCP server will reply with a DHCP-OFFER packet. This offer includes a client IP address, subnet mask, and default router.

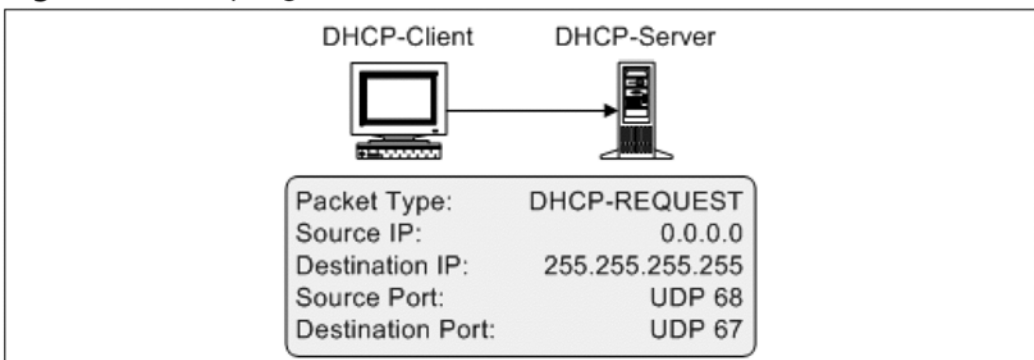
Figure 2.2 Receiving the DHCP Lease Offer(s)



Selecting and Accepting a DHCP Lease

If DHCP services are functioning properly, the client will receive at least one DHCP-OFFER. The client selects a lease offer (usually the first offer received) and sends a DHCP-REQUEST packet to the DHCP server. Although the client knows the IP address of the DHCP server, this packet is sent in broadcast mode so other DHCP servers will also know which DHCP-OFFER has been selected.

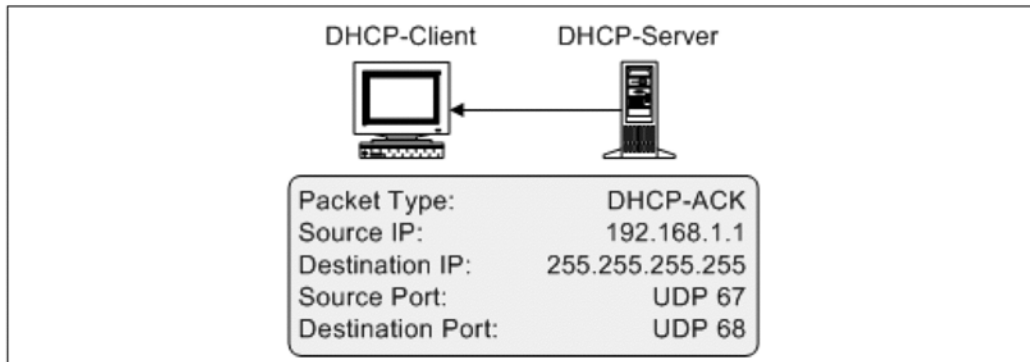
Figure 2.3 Accepting a DHCP Lease



Receiving Confirmation of DHCP Lease Assignment

Following the receipt of a DHCP-REQUEST packet, the DHCP server sends a DHCP-ACK packet confirming the DHCP lease assignment. This packet is sent in broadcast mode so other DHCP servers will know which DHCP-REQUEST packet has been confirmed.

Figure 2.4 Receiving Confirmation of DHCP Lease Assignment



With the receipt of the DHCP-ACK packet, the client is officially on the network with a valid IP address, subnet, and default router. Further DHCP communications (with the same DHCP server) will be made via unicast packets.

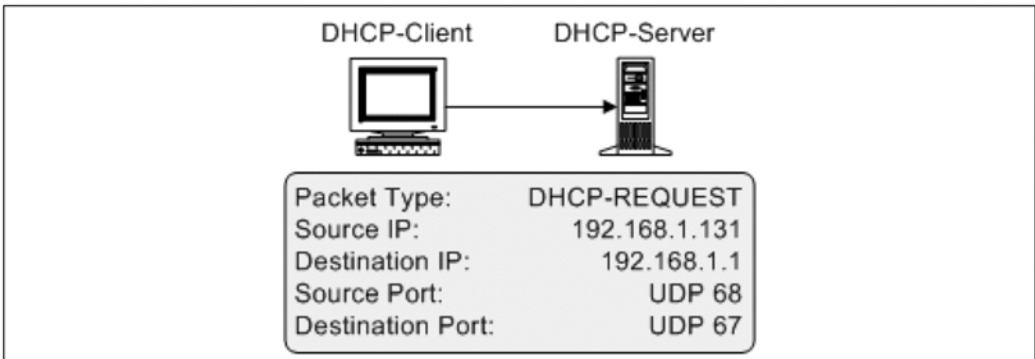
If there is a problem with the DHCP-REQUEST packet received from the client, the server will send a DHCP-NACK packet and wait for the client to retry.

Renewing a DHCP Lease

When 50% to 90% of the total lease time has elapsed, the DHCP client will attempt to renew the lease. The procedure to renew a lease is similar to the final two steps of the initial DHCP lease four-way handshake process.

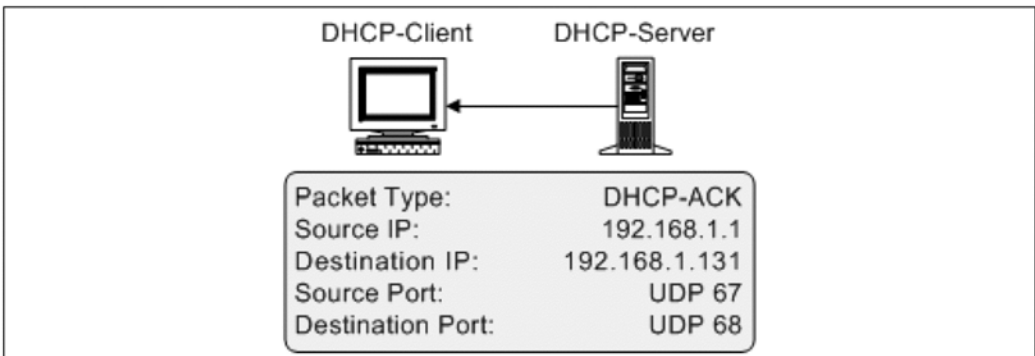
Requesting a DHCP Lease Renewal

Before the lease is set to expire, the client sends a DHCP-REQUEST packet to the DHCP server. This packet contains the client's current IP address settings.

Figure 2.5 Requesting a DHCP Lease Renewal

Receiving Confirmation of DHCP Lease Renewal

Following the receipt of a DHCP-REQUEST renewal packet, the DHCP server sends a DHCP-ACK packet confirming the DHCP lease renewal. If there is a problem with the DHCP-REQUEST packet received from the client, the server will send a DHCP-NACK packet.

Figure 2.6 Receiving Confirmation of DHCP Lease Renewal

Releasing a DHCP Lease

The final step in the DHCP communications process is the release of a DHCP lease. During the shutdown sequence of the network interface, the DHCP client sends a DHCP-RELEASE packet to the DHCP server. The DHCP server terminates the client's DHCP lease and returns the IP address to the pool of available IP addresses.

Examples and Exercises

Collecting and Analyzing DHCP Traffic

When analyzing a topic in-depth, there is no substitute for real-world data collection. The easiest way to collect DHCP traffic is to run **tcpdump** on the DHCP server. To collect DHCP packets on `eth0` and write them to the file `/dhcp-sniff`, enter the following command:

```
tcpdump -i eth0 -w /dhcp-sniff udp port 67 or udp port 68
```

After packet collection is complete, stop **tcpdump** by pressing **Ctrl + C**. Next, start **ethereal** with the following command:

```
ethereal /dhcp-sniff
```

This will browse and graphically analyze the DHCP data. For more information about **Ethereal**, consider reading *Ethereal Packet Sniffing* (Syngress).

Configuring DHCP Servers

Now that you understand how a DHCP client communicates with a DHCP server, let's take a closer look at the server configuration. We will use the Internet System Consortium's (ISC's) DHCPD www.isc.org/sw/dhcp/ to demonstrate how DHCP servers work. DHCPD is the world's most popular open source DHCP server, and is the DHCP server of choice on Linux.

DHCPD stores its configuration information in **dhcpd.conf**. Listed below are the significant portions of the post-migration `dhcpd.conf` files for Ballystyx's servers in Silicon Valley and India. We will discuss the **ddns-update-style** entry later in the chapter. This information may be used as a template when configuring your DHCP server. All time values are expressed in seconds.

```
ddns-update-style ad-hoc;
```

```
# Ballystyx Silicon Valley
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.130 192.168.1.250;
    default-lease-time 43200;
```

```
max-lease-time 86400;

option routers 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;

option domain-name "ballystyx.com";
option domain-name-servers 192.168.1.12, 192.168.2.10;
}

# Ballystyx India
subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.130 192.168.2.250;
    default-lease-time 43200;
    max-lease-time 86400;

    option routers 192.168.2.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.2.255;

    option domain-name "ballystyx.com";
    option domain-name-servers 192.168.2.10, 192.168.1.12;
}
```

Note that Ballystyx makes use of two DNS servers, 192.168.1.12 and 192.168.2.10. In Silicon Valley the primary DNS server is 192.168.1.12 and the secondary DNS server is 192.168.2.10. In India the situation is reversed. This ensures each DHCP client will first try to contact a local DNS server.

Security Tip

What is a Rogue DHCP Server?

While DHCP seems innocuous in its simplicity, there is a potential security flaw. What if a malicious hacker set up a rogue DHCP server on your network and fed bogus information to DHCP clients? Such bogus information could include such things as:

- **Invalid IP addresses**, such as those already in use elsewhere. This will cause IP address collisions and all kinds of network malfunctions and headaches.
- **Invalid routing information**, such as the IP address of a rogue router or routing process running on a hacker-controlled computer.
- **Invalid DNS information**, such as rogue or bogus DNS servers. This can disable a client's name resolution functionality, or make a rogue server appear to be a legitimate site.

As you can see, a rogue DHCP server can cause a large number of network headaches because so much relies on keeping IP addresses and leases properly sorted out. Fortunately, there are ways to defend against and/or prevent the problem of rogue DHCP servers:

- **Network Probing** Send various types of DHCP packets to locate DHCP servers on your network. If the responding DHCP servers do not match a list of known IP addresses (or another access control method of your choice), alert an administrator and / or shun the IP address.
- **Intrusion Detection Integration** Configure your intrusion detection engine to log UDP port 67/68 packets and alert for suspicious traffic.
- **Router and Switch Configuration** Set up restrictions on routers and switches regarding UDP port 67/68 packets.

Note that DHCP services can fail for a short while and the network will continue to function, particularly if no machines are moved and the lease expira-

tion time is sufficiently long. However, it is a good idea to have a monitoring process in place to alert administrators if DHCP services are not responding.

Understanding Name Resolution Services

As mentioned earlier in this chapter, the ability to translate computer host names from easily-remembered English words into IP addresses forms a core usability underpinning of all sizes and types of computer networks. People have a much easier time typing (and remembering) `www.google.com` than `216.239.39.99`.

Name resolution services are as old and varied as the Internet itself. Originally, all Internet hostnames were stored in a single text file on a single server. Obviously, this method did not scale well, as more and more bandwidth and computing power was required to deal with constant downloads of the text file. Today, there are two primary methods for managing hostname-to-IP translation services: file-based and DNS-based. We will examine both of these methodologies in detail in the following sections.

Understanding File-Based Name Resolution

The most simplistic method of name resolution is to store all names and corresponding IP addresses in a local file. In most cases the file is simply named `hosts`. The listing below contains the `hosts` file for Acme Widgets:

```
127.0.0.1      localhost      localhost.localdomain
192.168.100.1  router1        router1.acmewidgets.com
192.168.100.2  server1        server1.acmewidgets.com
192.168.100.3  admin1         admin1.acmewidgets.com
192.168.100.4  admin2         admin2.acmewidgets.com
192.168.100.5  widget1        widget1.acmewidgets.com
192.168.100.6  widget2        widget2.acmewidgets.com
192.168.100.7  printer1       printer1.acmewidgets.com
192.168.100.8  printer2       printer2.acmewidgets.com
```

As you can see, this provides all the basic name lookup capability required for a small LAN. However, there is no other functionality provided by this method, such as dynamic name registration or record expiration.

The use of a `hosts` file is common to Windows and Linux. Below are the usual locations of the file in various OS flavors.

Table 2.2 Hosts File Locations on Various Operating Systems

Operating System	Hosts File Location
Linux flavors	/etc/hosts
Windows 95 / 98 / Me	C:\Windows\hosts
Windows NT / 2000 / XP	C:\Winnt\system32\drivers\etc\hosts

In many cases, the Windows installation directory will be named WINNT. In other cases it will be named WINDOWS or another name chosen by the person who installed the operating system.

Understanding DNS Name Resolution

Although file-based name resolution can satisfy basic needs, it is too limited to be useful on anything but a very small network of static IP addresses. It does not provide the availability, scalability, or capabilities of a dynamic distributed system like DNS. DNS is the primary way of resolving names to IP addresses on both Windows (2000 and later) and Linux systems, as well as being the recognized name resolution mechanism for the Internet.

One of the reasons that DNS is so scalable is that the name resolution is split into domains (such as syngress.com), and the name resolution services for each domain can be split up among several machines to achieve redundancy and availability. DNS issues on one domain do not usually affect other domains. If a server does not know the answer to a DNS query, it will automatically refer the client to a DNS server that is authoritative for that domain.

There are 13 top-level DNS servers (called *root servers*) maintained by various organizations throughout the world. Approximately half of these servers are physically located in the United States. More information about these root servers can be found at www.root-servers.org.

In addition to the name-to-IP resolution services, DNS can provide IP-to-name resolution services. This reverse lookup feature is managed through the use of an in-addr.arpa domain and PTR (pointer) resource records. If a DNS client wants to know the name of a host with the IP address 192.168.1.12, the client will perform a PTR lookup on the record 12.1.168.192.in-addr.arpa. The DNS server that is authoritative for that subnet will return the name associated with that IP address. Reverse lookup is often used to verify the server's name, particularly in the case of sending e-mail. All Internet-accessible hosts should have a

PTR entry to allow reverse lookup queries to succeed. In-addr.arpa reverse-lookup zones are strongly recommended for internal DNS, too.

Understanding Dynamic DNS (DDNS)

In the early days of DNS, entries were managed by manually editing a text file (called the *zone file*) listing the names, IP addresses, and record types. In those days, computers rarely changed IP addresses and the information remained mostly static. Manually editing the zone file once in a while – adding/changing name and IP information and updating the serial number – was sufficient.

With the advent of dynamically-assigned IP addresses, manually editing DNS zone files was no longer a realistic option for busy network administrators. Some way of dynamically registering computer names was needed. This is the basis of RFC 2136 and *Dynamic DNS* (DDNS). Throughout the rest of this book, DNS will refer to DNS that includes Dynamic DNS.

In a DDNS environment, clients (or the DHCP servers) register their names and IP addresses with the authoritative name server for their domain, as defined by the Start of Authority (SOA) record. Clients also register a PTR record with the in-addr.arpa domain that is the SOA for their subnet.

Configuring BIND and DHCP for Dynamic DNS

In the default configuration, Windows 2000 and Windows XP clients automatically register themselves using Dynamic DNS after obtaining an IP address. The following entry in `dhcpd.conf` allows this to happen properly:

```
ddns-update-style ad-hoc;
```

If there are Windows 98 and NT clients on your network, they will not automatically register with Dynamic DNS. Instead, the DHCP server can be configured to register their hostname and a PTR/TXT resource record for their IP address.

```
ddns-update-style interim;
```

The interim DDNS update style is slightly different than the (to be ratified) RFC standards, but is effective in dealing with down-level clients. To find out more about this and other information about `dhcpd`, type **man dhcpd.conf** at a shell prompt.

To configure BIND to allow Dynamic DNS updates, ensure the following line is in `named.conf`:

```
allow-update;
```


TIP

Windows Internet Naming Service, or WINS, is an older naming service used in Microsoft networks, especially NT and Windows 3.x/95/98 networks. Outside of Microsoft networking, there are few uses for WINS, and even Microsoft has deprecated its use in favor of DNS. If you absolutely require a WINS server post-migration, Samba (covered later in this book) has this capability. However, in almost all circumstances, DNS is preferred and used exclusively by modern operating systems.

Understanding Time Synchronization Services

The importance of time synchronization cannot be overstated. It is of critical importance to ensure that the clocks of clients and servers are set to the correct time. If the time difference (called *clock skew*) is greater than five minutes, some services (such as Kerberos) will not function. Other services may function improperly, such as an e-mail server indicating that an e-mail was received 10 minutes before it was actually sent. In addition, it can be very difficult to troubleshoot problems with computers and network services if clock accuracy is not maintained. This is particularly true when trying to correlate log entries on multiple computers.

There are many ways (and protocols) to accomplish time synchronization, but the most popular and time-tested is the Network Time Protocol (NTP).

Understanding Network Time Protocol

For all modern operating systems, NTP is the protocol of choice to enable time synchronization between computers. NTPv3 is defined in RFC 1305. NTPv4 does not yet have a formal IETF RFC draft, but is implemented in the current NTP suite release. A subset of NTP, called Simple NTP (SNTP), defined in RFC 2030, may be used in cases where the network latency between the time server and client is minimal, as is the case in a corporate LAN. In the default configuration, Windows 2000/XP computers use SNTP to synchronize time with Windows servers.

NTP functionality is based upon the concept of master time servers (called *stratum 1* servers), which obtain the correct time from highly accurate sources such as locally attached Global Positioning System (GPS) or cesium clocks. A server that synchronizes with a stratum 1 server is called a stratum 2 server – the stratum of the source server plus 1. As the stratum number increases, the accuracy of the time can degrade slightly.

The principal problems with synchronizing time are accounting for network latency, packet processing time, and inaccurate time servers. For example, if a time server sends a packet that indicates, “The time is now exactly 12:00:00; set your clock to 12:00:00,” and the packet takes two seconds to reach the destination, the clock on the client computer would be two seconds slow. If the client requires one second to process the packet, the time on the client computer would then be three seconds slow.

NTP overcomes these problems in a number of ways:

- Measuring the network latency by using client and server timestamps.
- Accounting for the time required to process the network packets.
- Using multiple samples from multiple servers to ensure accuracy.
- Blacklisting servers that produce inconsistent or inaccurate results.

NTP uses UDP Port 123. More information about NTP can be found at www.ntp.org.

Understanding Linux NTP Clients

The most popular NTP client for Linux is the implementation by ntp.org. Ntp.org produces a complete client and server suite supporting NTPv3 and NTPv4. The suite includes:

- **ntpq** queries an NTP server.
- **ntpd** maintains accuracy of local clock and (optionally) provides NTP service to clients.
- **ntptrace** traces an NTP server chain to the source server.
- **ntpdate** one-time clock update program (soon to be deprecated).

Ntpd (the NTP daemon) can be run as an NTP client and/or server. If you configure one or more Linux servers as NTP servers, you can provide time synchronization services to your internal infrastructure. However, this is not always

as good an idea as it seems. If a malicious hacker (or an unskilled sysadmin) alters the clock on the time servers, the effect can propagate to all of the computers synchronizing with the time servers if there is no correct time data available.

A better option is to run the `ntpd` service on all Linux computers and set them up to synchronize with a list of stratum 2 and/or stratum 3 servers on the Internet. Although stratum 1 servers may be used for time synchronization, it is considered poor netiquette to utilize these servers unless you are providing stratum 2 time services to the Internet. Stratum 2 and 3 servers will provide sufficiently accurate time.

The file `ntpd.conf` controls the configuration settings for `ntpd`. The default configuration file is generally sufficient. If you would like to use the collection of NTP servers listed at pool.ntp.org, place the following line in `ntpd.conf`:

```
server pool.ntp.org
```

You can find out more information about `ntpd` and `ntpd.conf` by typing **man ntpd**. Go to www.pool.ntp.org for more information about the `ntp.org` time server pool.

Understanding Windows Time Service Clients

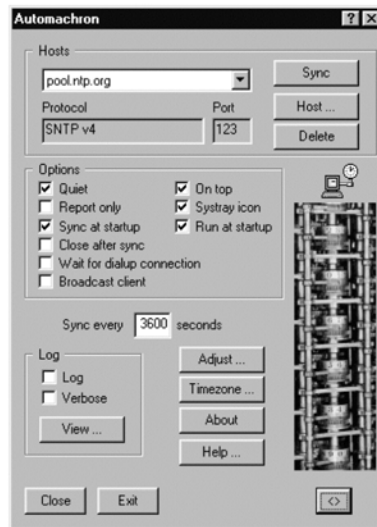
Windows 2000 and Windows XP ship with Microsoft's SNTP client, Windows Time Service. Windows Time Service is activated automatically when computers join a domain. For Windows computers that are not members of a domain, the service must be started manually.

In Windows 2000 and Windows XP, clocks may be synchronized with a `pool.ntp.org` time server using the following command:

```
net time /setsntp:pool.ntp.org
```

In many cases Windows will report, "The command completed successfully," but the time will not be set. In addition, this command does not work in Windows 98 or NT.

A better choice for Windows time synchronization is Automachron. Automachron is freeware and can be found at www.oneguycoding.com/automachron/. Unlike Windows Time Service, Automachron works with all versions of 32-bit Windows and features an easy-to-use GUI for configuration. The Automachron dialog box can be seen in Figure 2.7.

Figure 2.7 Automachron Configuration Dialog Box

If you find Automachron useful, visit www.oneguycoding.com and consider a PayPal donation.

Security Tip

NTP Traffic

In order to synchronize with Internet time servers, make sure that your firewall is configured to allow UDP port 123 traffic inbound and outbound. For security reasons, it is best to restrict NTP traffic only to the IP addresses of the NTP servers used by your organization.

Migrating MS-DNS/DHCP to Linux BIND/DHCPD

This migration section covers migration of DHCP and migration of DNS. Although these are two separate migrations, they must be performed in a coordinated fashion because of the interconnectedness of DHCP and DNS.

Migrating the DHCP Scopes and Options

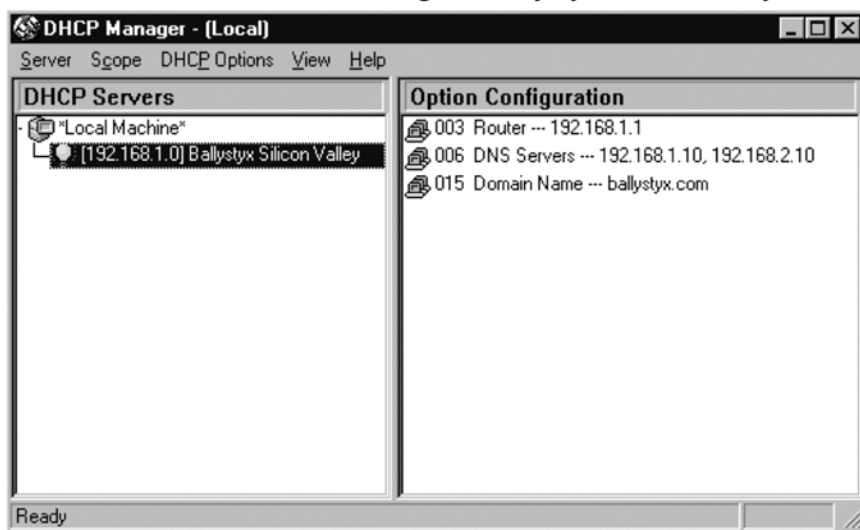
Migrating from Windows-based to Linux-based DHCP primarily consists of determining all of the scopes and important scope options, configuring DHCPD to use this information, shutting off Microsoft DHCP services, and starting Linux DHCP services.

Determining the DHCP Scopes and Options on Windows NT

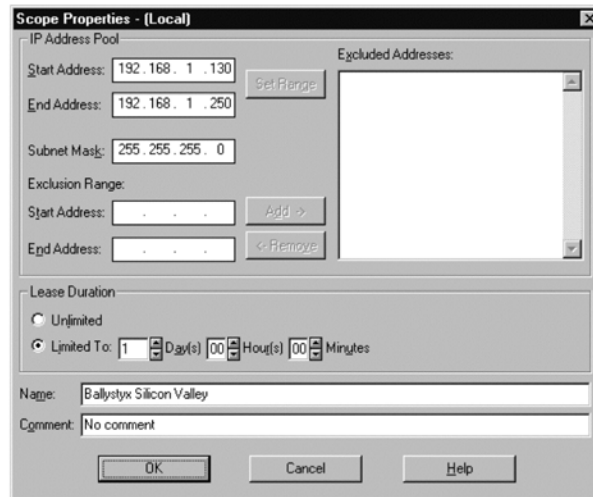
The first step in migrating DHCP services is determining all of the DHCP scopes serviced by MS-DHCP, and the detailed properties of each scope.

In Windows NT 4.0 Server, this is accomplished using DHCP Manager (dhcpcadm.exe). To access DHCP Manager (see Figure 2.8) using the mouse, select **Start | Programs | Administrative Tools | DHCP Manager**.

Figure 2.8 Windows NT DHCP Manager - Ballystyx Silicon Valley

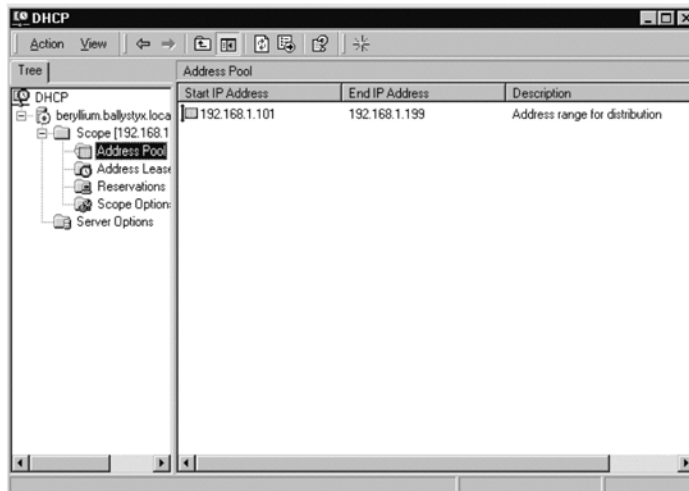


As you can see in the figure above, each scope that is serviced by this DHCP server is listed, along with the DHCP options for that scope. To obtain the scope property information (Figure 2.9), choose **Scope | Properties**.

Figure 2.9 Windows NT DHCP Manager - Scope Details

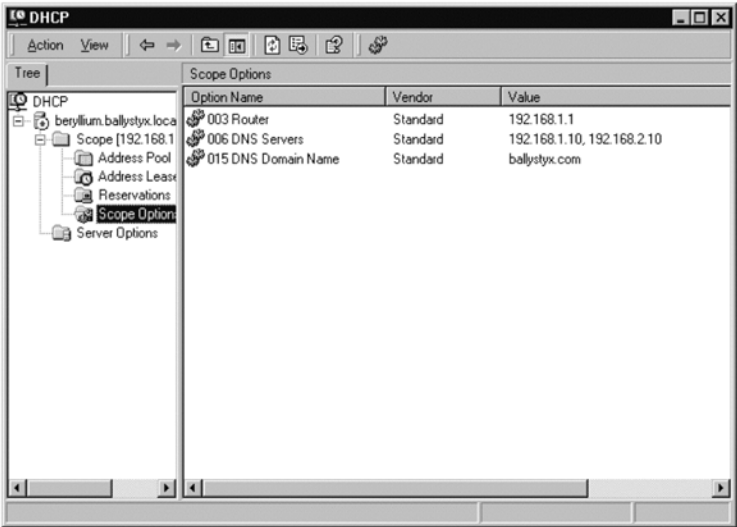
Determining the DHCP Scopes and Options on Windows 2000

To obtain the DHCP scope and IP address information in Windows 2000 server, select **Start | Programs | Administrative Tools | DHCP**. As you can see in Figure 2.10 below, each scope that is serviced by the DHCP server is listed, along with the DHCP Address Pool, Leases, Reservations, and Scope Options. Click on the Address Pool to obtain the range of IP addresses provided by this scope.

Figure 2.10 Windows 2000 DHCP Manager - Address Pool Range

To determine the scope option details, click on **Scope Options** (Figure 2.11).

Figure 2.11 Windows 2000 DHCP Manager - Scope Option Details



Recording the DHCP Scopes and Options

Record all of the DHCP information on paper or in a text file or spreadsheet. The minimum amount of information that you must gather is listed below in Table 2.3. If your version of DHCP server does not have a default lease time listed, you may set the default lease time to the maximum lease time or 50% of the maximum lease time.

Table 2.3 DHCP Scopes and Options Information

Component	Value(s)
Subnet	192.168.1.0
Netmask	255.255.255.0
Range	192.168.1.130 to 192.168.1.250
Default Lease Time	43200 seconds
Max Lease Time	86400 seconds
Router(s)	192.168.1.1
Subnet-mask	255.255.255.0

Continued

Table 2.3 DHCP Scopes and Options Information

Component	Value(s)
Broadcast Address	192.168.1.255
Domain Name	ballystyx.com
Domain Name Servers	192.168.1.12, 192.168.2.10

After all of the information for each DHCP scope has been collected and recorded, you are ready to configure the Linux DHCP infrastructure.

Configuring the ISC DHCPD Server

To set up Linux-based DHCP services, compile and install the current ISC DHCP server package or install the current DHCP server package from your Linux distribution on a test lab server. Start your favorite text editor and edit the `dhcpd.conf` file. For Ballystyx Silicon Valley, `dhcpd.conf` looks like this:

```
ddns-update-style ad-hoc;
# Ballystyx Silicon Valley
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.130 192.168.1.250;
    default-lease-time 43200;
    max-lease-time 86400;

    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;

    option domain-name "ballystyx.com";
    option domain-name-servers 192.168.1.12, 192.168.2.10;
}
```

Start DHCPD. Verify that a test lab workstation receives an IP address assignment and the appropriate DHCP options.

Migrating from Microsoft DHCP Services to Linux DHCP Services

Copy the known good `dhcpd.conf` file from the test lab server to your production Linux DHCP server. Stop the Microsoft DHCP services. Start the Linux DHCP services. Boot up a workstation and test DHCP functionality by ensuring that Windows desktops and other clients properly receive IP addresses and other DHCP information.

Backing Out of a Failed DHCP Migration

In some cases the new Linux DHCP services might not work properly, and workstations may be unable to receive IP addresses. If this happens, stop the Linux DHCP services and restart the Microsoft DHCP services. Repeat the tests in the lab environment and use the DHCP sniffing procedure (described above in a sidebar) to troubleshoot problems if necessary.

Migrating the DNS Information

There are many ways to migrate DNS information from MS-DNS to BIND. If there are only a small number of DNS entries, the migration can be performed manually by copying and pasting the information. DNS information may be obtained from Microsoft's DNS Manager or from the `.dns` files in `C:\WINNT\system32\dns\` and subdirectories. However, this method can be time consuming if there are many entries. For most organizations, the easiest way to transfer the information is using the DNS Zone Transfer (AXFR) mechanism.

Installing and Configuring BIND

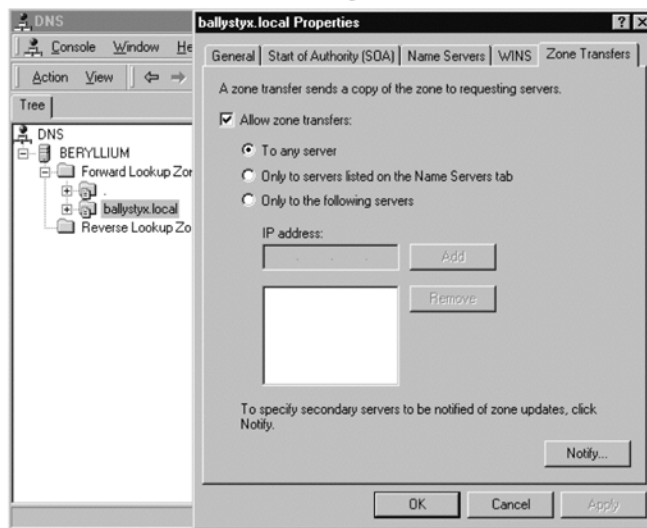
Compile and install the current ISC BIND package or install the current BIND server package from your Linux distribution on a test lab server. Start your favorite text editor and edit `named.conf`, the configuration file for BIND. The `ballystyx.com` entry in `named.conf` looks like this:

```
zone "ballystyx.com" {  
    type master;  
    file "/etc/bind/ballystyx.com.hosts";  
};
```

Transferring DNS Information

Most Windows DNS servers are configured to allow DNS Zone Transfers (AXFR) by default (dangerous for security, but useful for migrations!) so this option is usually available or can be made available. In Windows 2000, you can verify the settings by starting DNS Manager, expanding the **Forward Lookup Zones**, and right-clicking on the domain name and choosing **Properties**. Select the **Zone Transfers** tab and ensure that **Allow zone transfers** is checked as shown in Figure 2.12.

Figure 2.12 Windows 2000 DNS Manager Zone Transfers Dialog Box



The following command will perform a DNS zone transfer at Ballystyx:

```
dig @lithium ballystyx.com axfr
```

This will list all of the records in the ballystyx.com domain. The format is similar to the format for a BIND zone file. The output from **dig** looks like this:


```
; <<>> DiG 9.2.4rc5 <<>> @lithium ballystyx.com axfr
;; global options: printcmd
ballystyx.com.      3600    IN      SOA     lithium.ballystyx.com.
admin.ballystyx.com. 5 3600 600 86400 3600
ballystyx.com.      3600    IN      NS      lithium.ballystyx.com.
hydrogen.ballystyx.com. 3600    IN      A       192.168.1.10
helium.ballystyx.com. 3600    IN      A       192.168.1.11
```

```
lithium.ballystyx.com.      3600   IN      A       192.168.1.12
beryllium.ballystyx.com.   3600   IN      A       192.168.1.13
;; Query time: 2 msec
;; SERVER: lithium#53(192.168.1.12)
;; WHEN: Sat Jul 24 08:07:27 2004
;; XFR size: 6 records
```

You may manually copy the records and paste them into the BIND zone file, or use a graphical configuration tool such as Webmin to enter the data. This method works best if there are few DNS entries. When finished, the zone file looks like this:

```
ballystyx.com.IN      SOA      lithium.ballystyx.com. admin.ballystyx.com. (
                        5
                        3600
                        600
                        86400
                        3600 )


ballystyx.com.IN      NS       lithium.ballystyx.com.
hydrogen.ballystyx.com.  IN      A       192.168.1.10
helium.ballystyx.com.IN  A       192.168.1.11
lithium.ballystyx.com.  IN      A       192.168.1.12
beryllium.ballystyx.com. IN      A       192.168.1.13
```



Another method is to use the included migration script, `migrate-dns`, to automatically transfer the DNS information from Windows to Linux. This method is effective even for larger numbers of DNS entries.

Transfer the DNS information to a test lab BIND server and ensure that the information has been transferred correctly by looking at the zone file and performing DNS lookups.

Migrating from Microsoft DNS Services to Linux DNS Services



When you are sure that the Linux BIND server configuration is working in the lab, copy the known good `named.conf` file to your production Linux BIND server. Then perform a zone transfer manually or using the `migrate-dns` migration script. You have to modify the **SOA** and **NS** entries to reflect the new Linux DNS server if the name and/or IP address is different.

Boot up a workstation and test DNS functionality by ensuring that Windows desktops and other clients can properly translate names to IP addresses. If you are using DDNS to register DHCP clients, ensure that the client names are properly registered in DNS. If you are providing DNS reverse lookup capabilities, ensure that a PTR resource record is registered in the appropriate in-addr.arpa zone.

If you are using DHCP services, you must modify the **domain-name-servers** entry in `dhcpd.conf` to include the new Linux DNS server if the IP address is different. If it is feasible to reboot your workstations at this time, do so. This will ensure that they receive the updated DNS information, and will test all of the features you have migrated to Linux servers in this chapter. If rebooting is too disruptive, forcing a DHCP lease renewal is an acceptable option.

Backing Out of a Failed DNS Migration

In some cases the new Linux DNS services might not work properly, or DDNS name registration services may not function. If this happens, try changing the **ddns-update-style** entry. If functionality still does not work, stop the Linux DNS services and restart the Microsoft DNS services. Repeat the tests in the lab environment, and sniff port 53 traffic to troubleshoot problems if necessary. If you can determine and solve the cause of the problem, retry the migration.

Microsoft indicates that Active Directory will work properly with any DNS Server (such as BIND 9) that supports Dynamic DNS (RFC 2136) and SRV Resource Records (RFC 2782). However, real-world experience sometimes indicates otherwise, particularly in complex environments. If you are having problems with DHCP and DNS migration on Windows 2000 servers, you may have to wait until other MS-DNS-dependent services (such as Active Directory and Exchange 2000) have been retired before migrating DNS and DHCP to Linux.

Summary

IP address assignment, name resolution, and time synchronization services form the core services of any TCP/IP network, and enable the functioning of all other network services in this book. DHCP, DNS, and NTP are the protocols that provide these services to Windows and Linux clients. The functionality provided by these services is nearly identical when running on Windows or Linux, aside from the Active Directory integration of Windows 2000 DNS services.

The DHCP and DNS portion of the network services migration process is usually straightforward when migrating from Windows NT. Windows NT DHCP and DNS services are much less complex than Windows 2000 DHCP and DNS services, and are therefore easier to replace. Migration from Windows NT DNS and DHCP services to Linux-based DNS and DHCP services has the added benefit of enabling additional capabilities (such as DDNS) that are not present in Windows NT.

Migrating from Windows 2000 is usually straightforward in simple single-domain environments, but can sometimes be challenging in complex environments. If you were unable to migrate your Windows 2000 DNS and DHCP services to Linux because of technical problems, you can migrate these services after retiring MS-DNS dependent services like Active Directory and Exchange 2000 (which also depends on Active Directory).

Solutions Fast Track

Understanding IP Address Assignment Services

- ☑ There are two types of IP address assignment systems, static and dynamic. Most servers use static IP address assignments; most laptops and desktops use dynamic IP address assignments.
- ☑ DHCP is the protocol that dynamically assigns IP address and network configuration parameters to hosts.
- ☑ DHCP clients use a four-step broadcast process (a.k.a. four-way handshake) to obtain an initial DHCP lease. DHCP clients use a two-step unicast process to renew a DHCP lease.

- ☑ The Internet Systems Consortium (ISC) develops and maintains the world's most popular open source DHCP server and client software. Almost all versions of Linux utilize the ISC software for DHCP services. More information about ISC is available at www.isc.org.

Understanding Name Resolution Services

- ☑ The ability to translate computer host names into IP addresses forms a core usability underpinning of all sizes and types of computer networks.
- ☑ There are two types of name resolution methodologies: file-based and DNS-based. The use of a hosts file is appropriate only for very small static networks. All other networks use DNS for name resolution services.
- ☑ Dynamic DNS allows clients (or DHCP servers) to register their names and IP addresses with the authoritative name server for their domain, as defined by the SOA record. Clients (or DHCP servers) also register a PTR record with the in-addr.arpa domain that is the SOA for their subnet.

Understanding Time Synchronization Services

- ☑ The importance of time synchronization cannot be overstated. If the clocks of clients and servers are not set to the correct time, some network services may not function or may function improperly.
- ☑ For all modern operating systems, NTP is the protocol of choice to enable network time synchronization.
- ☑ Ntpd is the most popular time service client (and/or server) for Linux. Automachron is a popular time service client that runs on Windows 98, NT, 2000, and XP.
- ☑ Although it may be technically feasible to run an NTP time server on your internal network, a better option is to open UDP port 123 on your firewall and synchronize with low-stratum Internet time servers such as pool.ntp.org.

Migrating MS-DNS/DHCP to Linux BIND/DHCPD

- ☑ Although the migrations of DHCP and DNS services are two separate migrations, they must be performed in a coordinated fashion because of the interconnectedness of DHCP and DNS.
- ☑ Migrating from Windows-based DHCP services to Linux-based DHCP services primarily consists of recording all of the DHCP scopes and scope options, configuring DHCPD with this information, shutting down Windows DHCP services, and starting Linux DHCP services.
- ☑ Migrating from Windows-based DNS services to Linux-based DNS services primarily consists of installing a BIND server, collecting the DNS information via the zone transfer mechanism, and populating BIND with the information. The included migrate-dns script automates this process.
- ☑ If you are having problems with DHCP and DNS migration on Windows 2000 servers, you may have to wait until other MS-DNS-dependent services (such as Active Directory and Exchange 2000) have been retired before migrating DNS and DHCP to Linux.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Is there a graphical tool to configure DHCP and DNS services?

A: Yes! Webmin is a browser-based configuration tool that works with DHCP, DNS, and many of the other network services discussed in this book. Navigate to www.webmin.com for more information about Webmin and Webmin modules.

Q: Is DHCP the only method to manage dynamic IP address assignments?

A: OTP, the Bootstrap Protocol, is a much older dynamic IP address assignment system that provides only a subset of DHCP's rich capabilities. BOOTP is sometimes used in diskless workstations. Fortunately, `dhcpd` can service both DHCP and BOOTP clients.

Q: Is it possible to ensure that specific DHCP clients always receive the same IP address?

A: Yes. This is called a DHCP reservation, and is controlled via the Media Access Control (MAC) hardware address of the Network Interface Card (NIC) of the DHCP client. To accomplish this, add the following lines to **`dhcpd.conf`**, substituting the appropriate hostname, MAC, and IP address information for your computer(s):

```
host hostname.example.com {  
    hardware ethernet 00:11:22:33:44:55;  
    fixed-address 192.168.1.100;  
}
```


Q: Is there a way to allow only known DHCP clients to receive IP addresses?

A: Yes. This can be a good idea from a security perspective, although it slightly increases administrative overhead. To accomplish this, place the following entry in the appropriate section of **dhcpcd.conf**:

```
deny unknown clients;
```

Directory Services

Solutions in this Chapter:

- Understanding LDAP and Directories
- Understanding Microsoft Directory Services
- Understanding OpenLDAP
- Designing Linux-Based Directory Services

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Directory services form a crucial component for managing and cataloging objects such as user accounts and profile settings, groups, computers, printers, e-mail, and other network and infrastructure objects. Just as core network services are a prerequisite for directory services, directory services are a prerequisite (or a desirable integration point) for other network services such as authentication, messaging, and groupware services.

This chapter begins with an overview of directory services and Lightweight Directory Access Protocol (LDAP). The section covers LDAP fundamentals including Directory Information Tree (DIT) concepts, Distinguished Name (DN) convention, objectclasses, schema components, and other LDAP concepts. We also examine LDAP queries and connections.

Following the directory services overview, we explore Microsoft's directory services solutions. Microsoft's directory services changed considerably from Windows NT / Exchange 5.5 to Windows 2000. In Windows NT, the Security Accounts Manager (SAM) stores all user account information, and Exchange 5.5 provides extended contact and messaging information. In Windows 2000, both of these functions are combined into Active Directory (AD).

The next section examines OpenLDAP, the premier open source directory server. We examine features of the OpenLDAP suite including server daemons, client and server utilities, distributed directory services, and data import / export.

Directory services in a small company may consist of a single, non-dedicated server, whereas directory services in a larger company tend to be distributed, redundant, and utilized as an integration point for many other network services. The section entitled "Designing Linux-Based Directory Services" will help you determine key requirements for your company's directory services and design a flexible solution that meets the current requirements as well as allowing for future growth.

Understanding LDAP and Directories

LDAP is a cross-platform protocol that allows for communications with a directory server. LDAP is evolved from the Directory Access Protocol (DAP) component of an X.500 system. The primary functions of LDAP include authentication and query operations with an X.500-style directory. When we discuss LDAP in this book, we are referring to LDAPv3.

Understanding LDAP Terms

We begin our discussion of LDAP with an overview of the terms used to describe directory structure and directory objects. You need to understand this fundamental information in order to understand the rest of this chapter. For an in-depth analysis of LDAP, consult Adam Williams' *LDAP and OpenLDAP (on the Linux platform)* presentation at <ftp://kalamazoolinux.org/pub/pdf/ldapv3.pdf>.

Understanding Directory Structure

A directory is organized into a hierarchical structure. This structure is called the Directory Information Tree, or DIT. The DIT structure begins with the base DN (also called *suffix*) and is logically separated by organizational unit (ou) objects. DITs are typically organized by the type of object contained in each tree. In some situations, a company's geographical or business unit organization may affect DIT design.

Figure 3.1 shows a DIT for a typical company, such as Acme Widgets and Ballystyx. This type of DIT is usually sufficient for most companies.

Figure 3.1 Directory Information Tree – Simple Model

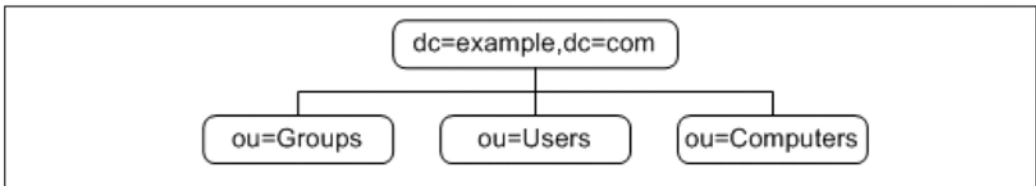


Figure 3.2 shows a DIT for a company with very clear delineation among business units. This might be a good way to organize a company where business units operate as separate “islands”, with almost no communication or information sharing between business groups.

Figure 3.2 Directory Information Tree – Business Group Model

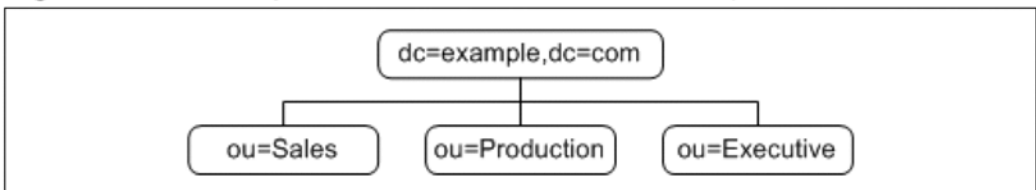
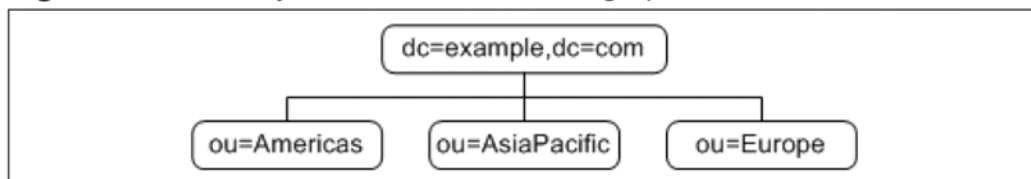


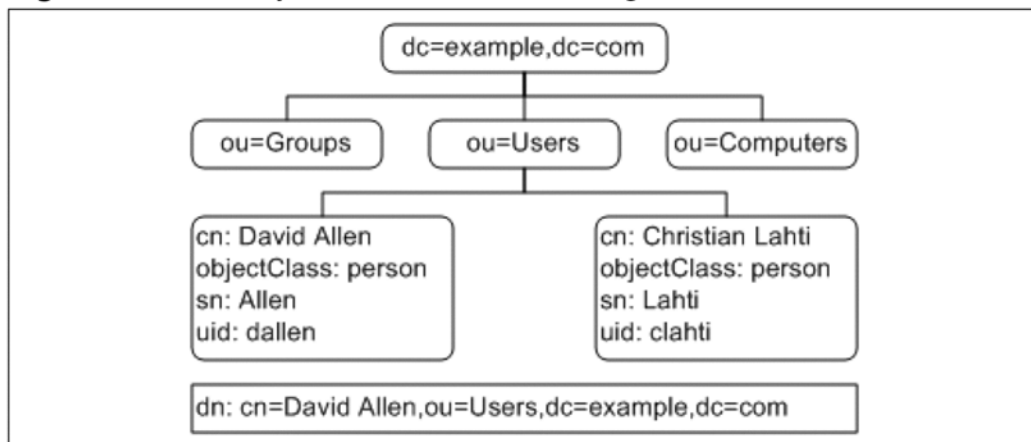
Figure 3.3 shows a DIT for a very large company with multiple large offices on multiple continents. In this case, all administrative, IT, and business functions are separated by continents for legal, financial, technical, and other business reasons. Although this type of organization is not applicable to most companies, it illustrates another type of directory structure and design.

Figure 3.3 Directory Information Tree – Geographical Model



Each object in the directory is uniquely identified by a distinguished name. The DN is composed of the attribute that uniquely identifies that object (usually common name (cn) or user ID (uid)) followed by the path that locates the object within the DIT. Figure 3.4 shows two example person objects and a corresponding DN.

Figure 3.4 Directory Information Tree – Distinguished Name



Understanding Directory Objects

A directory is a hierarchical store of objects and their attributes. Each object is an instance of one or more objectclasses. Objectclasses define the collection of attributes that may or must comprise these types of objects. Attributes may contain text, numeric, binary, or other types of data.

The collection of objectclasses supported by a directory server is called the directory schema. Schema files (such as those included with OpenLDAP and Samba) contain objectclass, attributes, and syntax definitions. Each of these entities is associated with an Object Identifier (OID). OIDs are represented by a hierarchical dotted-decimal notation, and uniquely identify object, attribute, and syntax definitions. The following shows the definition for the “person” objectclass in **core.schema**.

```
objectclass ( 2.5.6.6 NAME 'person'
    DESC 'RFC2256: a person'
    SUP top STRUCTURAL
    MUST ( sn $ cn )
    MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

This objectclass definition lists the attributes that *must* comprise a person object (common name and surname) and the attributes that *may* optionally be present (userPassword, telephoneNumber, seeAlso, and description). The following shows the definition of selected attributes.

```
attributetype ( 2.5.4.3 NAME ( 'cn' 'commonName' )
    DESC 'RFC2256: common name(s) for which the entity is known by'
    SUP name )

attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
    DESC 'RFC2256: last (family) name(s) for which the entity is known by'
    SUP name )

attributetype ( 2.5.4.20 NAME 'telephoneNumber'
    DESC 'RFC2256: Telephone Number'
    EQUALITY telephoneNumberMatch
    SUBSTR telephoneNumberSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.50{32} )
```

Connecting to a Directory Server

In order to perform operations such as LDAP queries, a client must first connect to the directory server. In LDAP parlance, this is called a *bind* operation. There are four types of bind operations:

- Anonymous bind
- Simple bind
- Simple bind with Transport Layer Security (TLS)
- Simple bind with Simple Authentication and Security Layer (SASL)

In an anonymous bind, the LDAP client supplies an empty binding DN and password. In an authenticated bind, the LDAP client supplies the DN of the directory object to bind as, and provides credentials such as a password. Simple binds (without encryption) have a disadvantage in that the password is transmitted in clear-text over the network.

In order to protect the password from sniffing, some form of encryption must be used. Most LDAP implementations support the use of the StartTLS operation, which encrypts all further communications between the client and the server. SASL is a more complex form of authentication that supports Kerberos and Generic Security Services Application Programming Interface (GSSAPI). We will discuss authentication and encryption in greater detail in Chapter 4. For now the most important detail to keep in mind is that non-anonymous binds that transmit clear-text passwords create a security issue.

Understanding LDAP Queries

Following a successful connection to the directory server, the LDAP client will typically issue a query. An LDAP query is composed of the following parts:

Search Base The search base contains the DN of the portion of directory tree where the search should begin. Typically this is in the form of `dc=domain,dc=com`, although older implementations may use `o=organization,c=country`. This is also sometimes called the search suffix.

Search Scope The search scope may be **base**, **one**, or **sub**. A base search scope searches only at the search base level. A search scope of **one** searches up to one level below the base. The sub search scope will search through all subtrees below and including the search base.

Search Filter The search filter specifies which types of objects and attributes the directory server should return. Search filters are specified using regular expression syntax.

Search Attributes The search attributes parameter lists the attributes to return in the LDAP query. If the search attribute field is not popu-

lated, all attributes of the object will be returned. To minimize the performance impact to the directory server, it is considered a best practice to ask for only the attributes that are needed.

Another way to represent an LDAP search is through the use of a Uniform Resource Indicator, or URI. The basic format of an LDAP URI is as follows:

```
'ldap://' [hostport] ['/' [dn ['?' [attributes] ['?' [scope] ['?' [filter]]]]]]
```

More information about LDAP search filters can be found in RFC 2254, “The String Representation of LDAP Search Filters.” More information about LDAP URI format can be found in RFC 2255, “The LDAP URL Format.”

Understanding Microsoft Directory Services

Windows NT and Windows 2000 Server employ three directory servers. These three servers are the NT Security Accounts Manager, Exchange 5.5 Directory Services, and Windows 2000 Active Directory. We examine these directory servers below.

Understanding Windows NT SAM

Windows NT makes use of a SAM to store user objects. The Windows NT SAM is managed by User Manager or third-party utilities and contains only a limited amount of information compared to other directory services. There is no LDAP interface to NT SAM.

Windows NT SAM also supports the use of groups, including nested groups. The SAM splits groups into two types, local and global. Local groups can contain global groups, but global groups cannot contain local groups.

The Windows NT SAM stores usernames and computer accounts internally as a Security Identifier (SID). A user account also contains LANMAN (LAN Manager) and NTLM (NT LANMAN) password hashes, the full user name, description, home directory, profile path, logon script, logon times, and account status.

Windows NT is a single-master directory service. A master (read-write) directory server is called a Primary Domain Controller (PDC), and a slave (read-only) directory server is called a Backup Domain Controller (BDC). There is

only one PDC in a Windows NT domain, but there may be zero or more BDCs.

Windows NT domains are not hierarchical, but instead depend on trust relationships for interoperation. In this book, we shall consider only a single-domain model. If you have multiple domains, you may use the single-domain migration techniques to migrate some or all portions of each domain individually. These domains may be kept separate or collapsed into a single domain, depending on your requirements.

Understanding Exchange 5.5 Directory Services

Exchange Server was the first implementation of Microsoft directory services to feature multi-master replication and LDAP support. The Exchange 5.5 directory server supports LDAPv3 and a rich set of objects and attributes. Each Exchange server contains a full copy of the directory. Changes to the Exchange directory made on one Exchange server propagate to all other Exchange servers within the Exchange organization.

Exchange supports the use of groups to create mailing lists. In Exchange 5.5, these groups are called *Distribution Lists*. However, these groups are not the same as NT SAM groups and are not interchangeable. Part of the advantage of Active Directory over the Exchange 5.5 directory is that AD groups may be used for both e-mail and access control.

Exchange 5.5 supports the concept of external recipients. The recipients are called *Custom Recipients*, and are generally accessible to all persons using the Exchange directory. They are similar to contacts in Outlook and other e-mail clients, but exist in the directory instead of on the local mail client.

Exchange 5.5 links NT SAM user objects to Exchange user mailboxes via the Primary Windows NT Account (Assoc-NT-Account) field. In Exchange Administrator, this field shows up as a Windows NT account, although it is actually stored as a SID in the Exchange directory.

The Exchange directory is a precursor to Active Directory, and contains a subset of Active Directory features and functionality.

Understanding Active Directory

Active Directory is Microsoft's Windows 2000 implementation of directory services. Active Directory is used as the directory server for Microsoft enterprise domain implementations, and forms a core requirement for many of Microsoft's

products, including Exchange Server. Active Directory runs only on Windows servers.

Active Directory implements an LDAP interface, although Active Directory Services Interface (ADSI) is the most widely used method to access Active Directory objects. Active Directory is tightly integrated with Microsoft's Windows 2000 DNS (Domain Name System) and DHCP (Dynamic Host Control Protocol) implementation, and requires DNS services in order to function properly.

While Active Directory performs similar basic functions to the NT Domain model, there are a number of differences between these two directory services. Table 3.1 summarizes these differences.

Table 3.1 Differences Between Windows NT Domains and Active Directory Domains

Active Directory Domains	Windows NT Domains
DNS-integrated.	No DNS integration.
Hierarchical, transitive trust model.	Island model with various types of trust relationships.
Store credentials for Kerberos and NT/LANMAN authentication.	Stores credentials for NT/LANMAN authentication. No Kerberos capabilities.
Stores extensive user contact information.	Stores very limited user contact information.
Multi-master replication model.	Single-master replication model.

Active Directory stores a considerable amount of information, and is a superset of the NT SAM and Exchange 5.5 directory. Username, full name, description, password hashes, home directory, profile path, contact information, e-mail address(es), and other information are stored in Active Directory. More information about Active Directory is available at: www.microsoft.com/windows2000/technologies/directory/

Understanding OpenLDAP

OpenLDAP is the premier open source directory solution. It contains a full suite of client and server components necessary to implement and utilize directory services. OpenLDAP runs on all versions of Linux and features robust replication

support for distributed directory services. OpenLDAP is a mature product. It was originally developed by the University of Michigan, but is now managed by the OpenLDAP Foundation. More information about OpenLDAP is available at www.openldap.org.

Understanding OpenLDAP Server Daemons

Slapd (Stand-alone LDAP Daemon) is the OpenLDAP server process that listens on network ports and responds to LDAP connections from clients. Slapd is managed through the use of the `slapd.conf` file, usually found in `/etc/openldap` or `/etc/ldap`.

Slurpd (Stand-alone Update Replication Daemon) enables distributed directory services by providing replication capabilities. Slurpd is capable of master-slave (single-master) or master-master (multi-master) models of replication. Master-slave replication works by maintaining a single read-write copy (the master) of the directory, and replicating the directory to one or more read-only copies (the slaves). Single-master replication is the least complicated replication method to use, and is the method used by most organizations. Slurpd uses a push style of replication, meaning that the master initiates connections to the other servers and pushes updates to them.

Multi-master replication works by allowing multiple directory servers to write to the directory. When a change occurs, the directory server processing the write operation sends an update to the other directory servers.

Understanding OpenLDAP Utilities

The OpenLDAP suite features a number of client utilities and libraries. Table 3.2 lists commonly used OpenLDAP utilities.

Table 3.2 Commonly-Used OpenLDAP Utilities

OpenLDAP Utility Name	Description
ldapsearch	Queries a directory server
ldapadd	Adds a directory object
ldapmodify	Modifies a directory object
ldapdelete	Deletes a directory object
ldappasswd	Changes the password of a directory object
slappasswd	Generates encrypted passwords

Continued

Table 3.2 Commonly-Used OpenLDAP Utilities

OpenLDAP Utility Name	Description
slapcat	Writes contents of directory to LDIF file
slapadd	Adds contents of a directory from LDIF file generated by slapcat
slapindex	Regenerates slapd indices

In order to maintain database consistency, slapd should be shut down (or running in read-only mode) when performing a **slapcat** or **slapindex** operation. Slapd should also be shut down during a **slapadd** operation. In most cases, an LDIF file is specified for input when executing **ldapadd**, **ldapmodify**, or **ldapdelete**, especially if multiple objects are to be affected. More information about OpenLDAP utilities may be obtained by typing **man utilityname**.

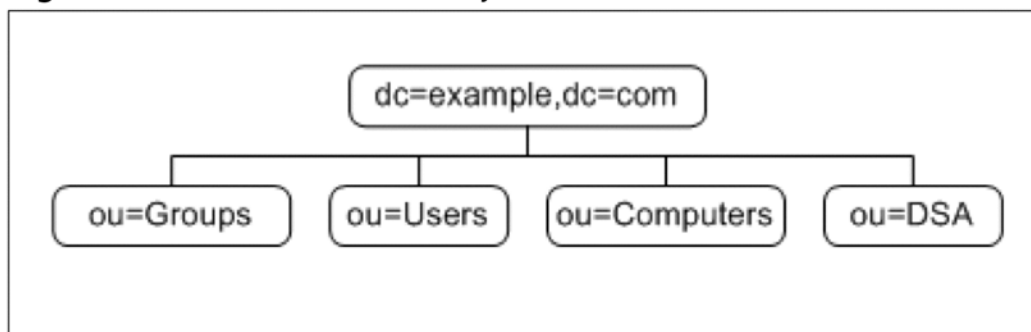
Designing Linux-Based Directory Services

With an understanding of directory services, LDAP, and OpenLDAP, you are ready to design and deploy your Linux-based directory services. The following sections will help you determine the best way to meet your organization's directory services requirements.

Designing a Directory Information Tree

The first step in directory services design is determining the structure of your organization's DIT. This process starts with determining your base DN. For a company with a .com Internet domain name, this usually means using **dc=your-domain,dc=com** as the base DN. For a company without an Internet domain name, you may consider using **dc=companyname,dc=local**.

After determining your base DN, you must determine how to logically separate your directory hierarchy into organizational units. For the vast majority of companies, only four organizational units are needed. Figure 3.5 illustrates the recommend DIT structure.

Figure 3.5 Recommended Directory Information Tree Structure

One of the advantages of this DIT structure is that it seamlessly allows you to use the Webmin www.webmin.com modules created by IDEALX for LDAP administration of Samba and Posix accounts. Using the `idxldapaccounts` modules is a recommended way for managing your directory. Information and downloads are available at www.idealx.org/prj/webmin/index.en.html.

If your organization is large and there is negligible communication and/or resource sharing between business units and/or office locations, you may consider a more complex DIT structure. However, in most cases it is best to use the recommended DIT, adding additional top-level organizational units as needed.

Designing the OpenLDAP Infrastructure

Now that you have determined the structure of your DIT, you must determine the type, placement, and number of OpenLDAP servers in your organization. For a very small company like Acme Widgets with few employees and only one server, the choice is simple: Acme Widgets installs OpenLDAP on the only available server.

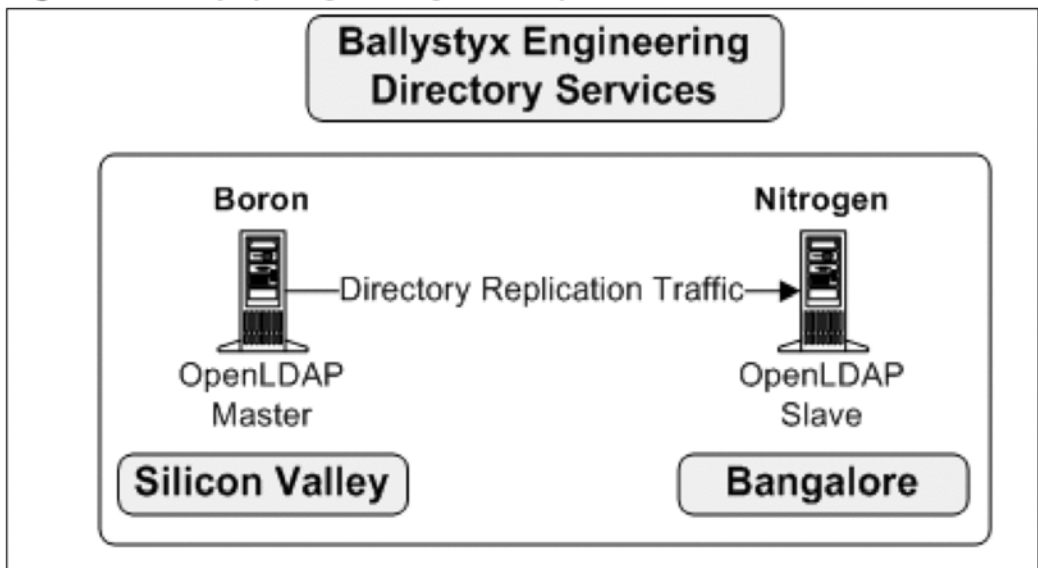
For a company like Ballystyx with many employees, multiple sites, and five servers, more options are available. The following criteria help to shape the decision-making process in medium- and large-sized organizations:

- Bandwidth usage and availability
- Server availability and scalability
- Directory service usage patterns including:
 - Usage patterns of end-user clients (such as e-mail address lookups)

- Usage patterns of server clients (such as authentication and messaging)
- Usage patterns at each site, and number of sites requiring directory service
- Frequency and type of directory read and write operations
- Directory replication and disaster recovery considerations
- Security and legal considerations
- Support personnel availability

Given these considerations, the directory services infrastructure design that makes the most sense for Ballystyx is to use single-master replication with the master OpenLDAP server located in Silicon Valley and a slave OpenLDAP server located in the Bangalore office. Figure 3.6 illustrates this arrangement.

Figure 3.6 Ballystyx Engineering Directory Services



Designing & Planning

Mission-Critical Directory Services

In an organization like Ballystyx Global Semiconductor Engineering, directory services are mission-critical and must be available at all times. One of the reasons for this is that other mission-critical network services depend on directory services. Authentication, messaging, and groupware services require directory services to function.

In order to accomplish this, there are two directory servers at Ballystyx – one in Silicon Valley and one in Bangalore. This ensures that another directory server is available if one server fails, as long as the network connectivity between Silicon Valley and India is functional. As budget is available and Ballystyx adds employees, additional OpenLDAP servers will be deployed in Silicon Valley and Bangalore to provide fully redundant load-balanced directory services at both locations.

Configuring and Testing the OpenLDAP Server(s)

With the type and placement of your server(s) determined, you are ready to test directory services in the lab. Compile and install the current OpenLDAP package or install the current OpenLDAP package from your distribution on a test lab server. Depending on which features of OpenLDAP you wish to use, you may need to supply switches to **./configure** to enable specific compile-time options. If you are installing an OpenLDAP package from a distribution, there may be multiple similarly named versions compiled with different options. For basic unencrypted directory services, the default options should work well. If you need additional features, you may recompile or upgrade the OpenLDAP software at a later time. More information about building and installing OpenLDAP is available at www.openldap.org/doc/admin22/install.html.

After you install OpenLDAP, you must customize the OpenLDAP server configuration file, `slapd.conf`, to reflect your environment. The following illustrates the `slapd.conf` file for Acme Widgets.

```
# Schema files
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/samba.schema

schemacheck on
lastmod on

# This section defines the acmewidgets.com directory database
database bdb
suffix "dc=acmewidgets,dc=com"
rootdn "cn=Manager,dc=acmewidgets,dc=com"
rootpw "secret"
directory /var/lib/ldap

# These database indices improve performance
index      objectClass,uidNumber,gidNumber          eq
index      cn,sn,uid,displayName                     pres,sub,eq
index      mail,givenname,memberUid                 eq,subinitial
index      sambaSID,sambaPrimaryGroupSID,sambaDomainName eq

# Access Control
access to attrs=userPassword,sambaNTPassword,sambaLMPassword
        by self write
        by anonymous auth
        by * none
# all others attributes are readable to everybody
access to *
        by * read
```

Note the included schema definitions. In many cases you must copy **samba.schema** from the Samba package. These schemas are required to provide the following objectclasses that will be used for the migration:

- top
- posixAccount
- shadowAccount
- sambaAccount
- sambaSAMAccount
- sambaDomain
- person
- organizationalPerson
- inetOrgPerson
- posixGroup
- sambaGroupMapping

When you have properly set up `slapd.conf`, start the OpenLDAP server and perform a simple bind operation using the rootdn credentials (`cn=Manager,dc=yourdomain,dc=com`). Although there will be no visible data other than the schema information, you have verified that you are able to properly connect to the directory server. You are now ready to populate the directory with your data.

Summary

Directory services enable the hierarchical storage and centralized management of information about users, groups, computers, and other network and infrastructure items. Since directory services are a prerequisite (or a desirable integration point) for the other network services mentioned in this book, deploying your Linux-based directory services represents the establishment of a significant foundation for the network services in the following chapters.

In the next chapter you will be introduced to Samba, which will integrate with your OpenLDAP directory to allow for authentication. We will also examine the migration of NT, Exchange 5.5, and/or Active Directory information.

Solutions Fast Track

Understanding LDAP and Directories

- ☑ A directory is a hierarchical store of objects and their attributes. Lightweight Directory Access Protocol (LDAP) allows for authentication and query operations with a directory server.
- ☑ The hierarchical structure of a directory is called a Directory Information Tree, or DIT. The DIT begins with a base DN, or directory suffix, such as `dc=example,dc=com`. Organization unit (ou) objects are used to create the tree-like hierarchical structure. Directories trees are usually organized by object type, but may also be organized by geography or business unit structure.
- ☑ The schema defines the objectclasses and attributes supported by a directory. An objectclass definition lists the attributes that must or may comprise an object of that type. Objectclasses and attributes are referenced by a unique Object Identifier (OID).
- ☑ Prior to performing a query, an LDAP client binds with the directory server. Binding may be anonymous or authenticated.

- ☑ LDAP queries are composed of a search base, scope, filter, and/or attribute list. An LDAP query may be defined by command-line arguments or represented by a Uniform Resource Indicator (URI). More information about LDAP search filters and URIs can be found in RFCs 2254 and 2255.

Understanding Microsoft Directory Services

- ☑ In Windows NT and Windows 2000 server there are three types of directory services: NT Security Accounts Manager (SAM), Exchange 5.5 directory services, and Active Directory.
- ☑ The Windows NT SAM is a very simple directory service that stores information about user accounts, computer accounts, and groups. Windows NT domains contain a Primary Domain Controller (PDC) and zero or more Backup Domain Controllers (BDCs). NT domains utilize single-master replication with the PDC having a read-write copy of the directory and BDC(s) having a read-only copy.
- ☑ The Exchange 5.5 directory features multi-master replication and LDAP support, and contains a rich set of objects and attributes. The Exchange directory supports user mailboxes, groups (for mailing lists), and custom recipients. User mailboxes are linked to the NT SAM through the use of a **Primary Windows NT Account** field.
- ☑ Active Directory is Microsoft's Windows 2000 implementation of enterprise directory services. Active Directory contains a superset of the information in the NT SAM and Exchange 5.5 directories. More information about Active Directory is available at www.microsoft.com/windows2000/technologies/directory/.

Understanding OpenLDAP

- ☑ OpenLDAP is the premier open source directory solution. It contains a full-featured suite of client and server components necessary to implement and utilize directory services.
- ☑ Slapd (Stand-alone LDAP daemon) is the OpenLDAP server process that responds to LDAP connections from clients. Slurpd (Stand-alone

Update Replication daemon) enables OpenLDAP distributed directory services by providing replication capabilities.

- ☑ The OpenLDAP suite contains a number of utilities, including OpenLDAP `ldapsearch`, `ldapadd`, `ldapmodify`, `ldapdelete`, `ldappasswd`, `slapasswd`, `slapcat`, `slapadd`, and `slapindex`. More information about OpenLDAP utilities may be obtained by typing **man *utilityname***.

Designing Linux-Based Directory Services

- ☑ The first step in designing Linux-based directory services is determining the structure of your organization's DIT and determining the base DN (directory suffix). For a company with a .com Internet domain name, this usually means using `dc=yourdomain,dc=com`.
- ☑ The next step is to determine the type, placement, and number of OpenLDAP servers in the organization. Bandwidth, server availability/scalability, directory usage patterns, replication, disaster recovery, security, and support considerations shape this decision-making process.
- ☑ The final step prior to deployment is to install, configure, and test directory services in the lab. Customize `slapd.conf` to include the schema files, base DN, database configuration, and access controls appropriate for your organization.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- Q:** Is there an easy way to back up or restore the contents of an OpenLDAP directory?
- A:** The use of slapcat and slapadd is recommended for these purposes. Do not attempt to copy the database file of a live database that is performing write operations – the database may be in an inconsistent state and cause a restore operation to fail.
- Q:** Why is the Windows SAM considered to be a directory server?
- A:** Although the Windows SAM is not an X.500-style directory and lacks an LDAP interface, it is a storage and query mechanism for users and group objects. For this reason it is grouped with other Microsoft directory servers.
- Q:** Are one, two, or three OpenLDAP servers enough for my organization? What is the best way to scale directory services, and what are the typical bottlenecks?
- A:** In an organization with well-designed distributed directory services, there is usually one read-write (master) OpenLDAP server at headquarters, and a read-only server at each branch office. If certain applications (such as postfix or Samba servers) perform a significant amount of directory lookups, a dedicated OpenLDAP server (read-only) may be deployed for these applications.
- Q:** Is there a simple way to redirect clients to another directory server if one fails?
- A:** Using round-robin DNS is a best-practices methodology to enable load balancing, redundant, and highly available directory services.

Authentication Services

Solutions in this Chapter:

- Understanding Windows Authentication
- Understanding Linux Authentication
- Designing Linux-Based Authentication Services
- Migrating from NT / Exchange or Active Directory

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Authentication is one of the most important and widely used network services. The ability to control access to network resources is primarily dependent on being able to reliably determine WHO you are. In most cases, authentication of users consists of validating credentials consisting of one or more factors: something you know (username, password), something you have (token), or something you are (biometric).

Authentication can be as simple as matching as typed password to a clear-text password stored in a local file. However, the security, scalability, reliability, and feature set requirements of all but the smallest companies mean utilizing an authentication scheme featuring an authentication service that interacts with a directory service (which may or may not be running on the same machine). Authentication credentials (such as usernames and password hashes) are stored in the directory, not within the authentication service. With the directory acting as a centralized repository of this information, you can build flexible authentication methodologies that allow for centralized control of distributed heterogeneous authentication services.

To an end user at a Windows workstation, one of the most visible examples of authentication occurs when the user types a username and password and logs in to the workstation. “Understanding Windows Authentication” examines authentication, logon script, drive mapping, and other processes that take place during login to a Windows workstation.

The next section, “Understanding Linux Authentication,” examines numerous types of authentication including `/etc/passwd/shadow`, LDAP (Lightweight Directory Access Protocol), and NIS (Network Information System). The section ends with an overview of Pluggable Authentication Modules (PAM), a flexible methodology-independent authentication mechanism.

“Designing Linux-Based Authentication Services” will help you to determine key authentication requirements based on your organization’s security, geography, OS (Operating System) support, and other requirements. We examine the authentication services provided by Samba and OpenLDAP, and guide you through the steps of installing Samba and integrating your installation with OpenLDAP.

The final section walks you through migrating information from NT / Exchange 5.5 / Active Directory (AD) into OpenLDAP using the supplied migration scripts. This section covers the migration of user, group, and computer information.

Understanding Windows Authentication

Beginning with Windows NT 3.1, Microsoft has utilized an implementation of the LANMAN protocol (NT LAN Manager, or NTLM, is the current incarnation) for authentication of user logins. Microsoft's implementation of the LANMAN protocol is derived from IBM's implementation of LANMAN in OS/2.

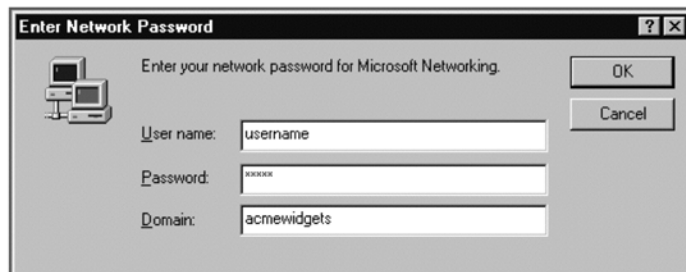
Microsoft introduced a Kerberos-based authentication mechanism in Windows 2000. Utilizing Kerberos for authentication enabled Microsoft's infrastructure to be more secure, scalable, and manageable by larger companies, and the integration with Active Directory simplified administration of what is considered a complex authentication protocol.

Microsoft's Kerberos implementation interoperates with – or has replaced – many servers running open source or non-Microsoft Kerberos services. One of the interoperability problems with Active Directory is that Microsoft “embraced and extended” the Kerberos authentication protocol to include the use of a cryptographically signed authorization component in the Kerberos PAC (Proxy Auto Configuration) field. While this is not an issue for Linux clients, Windows 2000/XP clients expect and require a populated PAC field. Although a Microsoft Kerberos server can authenticate Windows and non-Windows Kerberos clients, a Heimdal or MIT Linux Kerberos server will not properly authenticate Windows clients. Third-party Windows Kerberos clients are available that don't have this limitation (and therefore may authenticate using a Linux Kerberos server), but some clients (such as the one available from NCSA) are restricted by U.S. export laws due to the use of strong encryption.

Understanding Windows 98/NT Logon

Every morning, millions of home and small business users walk up to their computer, turn on the monitor, and see the screen shown in Figure 4.1.

Figure 4.1 Windows 98 Logon Screen



The user types in a username (usually present from the last login), a password, and the domain (also usually present), and clicks **OK**. The authentication subsystem is invoked for a domain-style (not locally-authenticated) logon. Utilizing the name resolution system, a domain controller is found and contacted. Windows 98 utilizes the LANMAN protocol, and Windows NT utilizes the NTLM protocol to pass the username, password hash, and domain credentials to the authentication server. The authentication server (a primary domain controller – PDC or backup domain controller – BDC in Windows NT parlance) parses the information and attempts to match up the credentials to those present in the SAM (Security Accounts Manager) database. If the credentials are valid, an authentication token is passed to the client. If the credentials are invalid, an error message is returned. If the user has a roaming profile configured, the copy on the server will be downloaded if it is newer than the local profile. If a user logon script is configured, this file will also be executed. Logon scripts are typically accessed via the NETLOGON share.

Understanding Windows 2000/XP Logon

The logon process for Windows 2000/XP is similar to the Windows 98/NT process. However, Windows 2000 and Windows XP attempt Kerberos authentication by default, although they also are backward compatible with NTLM authentication. If NTLM authentication is being attempted, the authentication process is nearly identical to the NT logon process, described above.

Windows 2000 Active Directory allows Group Policy Objects (GPOs) to apply configuration settings and launch (or install) applications on Windows 2000/XP desktops. At the time of this writing, the management / emulation of GPOs is not available for open source directory implementations. If you depend on GPO settings for your Windows 2000/XP desktops, you must utilize another method of applying settings and launching applications, such as moving these features to the logon script.

Understanding Linux Authentication

Historically, forms of UNIX authentication have existed for over forty years. From the venerable `/etc/passwd` to modern-day PAM, various methods of authentication have been utilized for *nix workstation logon. This section will examine authentication methodologies that function across the many flavors of Linux. In many cases, this information is also applicable to Solaris, BSD, MacOS, and HP-UX.

Understanding LDAP Authentication

LDAP authentication is one of the most widely used cross-platform enterprise authentication technologies. It is versatile, well supported, and features strong encryption. As mentioned above, authentication services frequently integrate with directory services. In the case of LDAP authentication, the authentication service is the directory service. In small companies with simple authentication needs, OpenLDAP is often used as the authentication server for *nix systems.

While LDAP authentication is elegant in its simplicity, it has drawbacks when used as the sole authentication service. LDAP was never designed exclusively to be an enterprise authentication protocol, as is the case with LANMAN and Kerberos. It does not employ the same authentication tokens or security components. Therefore, seamless single-sign-on authentication is difficult to achieve using only LDAP simple bind authentication.

Fortunately, the utility of LDAP authentication can be increased substantially through the use of SASL (Simple Authentication and Security Layer) and TLS/SSL (Transport Layer Security / Secure Sockets Layer). SASL provides a rich set of authentication technologies, and TLS/SSL enables encryption to ensure data and password (hash) privacy. The following authentication mechanisms supported by SASL:

- ANONYMOUS
- CRAM-MD5
- PLAIN
- DIGEST-MD5
- GSSAPI

Part of the appeal of GSSAPI (Generic Security Services Application Programming Interface) is that it further extends the authentication capabilities of SASL with a rich API (Application Programming Interface). GSSAPI also enables the use of Kerberos as an authentication mechanism. SASL was developed at Carnegie-Mellon University (CMU), and more information about CMU's Cyrus SASL implementation is available at <http://asg.web.cmu.edu/sasl/>.

Another important function of LDAP authentication is to serve as *lowest common denominator* authentication services. Since almost all authentication services can utilize LDAP to communicate with a directory server that stores username/password combinations, new authentication services are often easy to deploy when there is an existing LDAP server with authentication support enabled.

Understanding /etc/passwd/shadow Authentication

One of the simplest ways of managing authentication is to use a local file containing usernames and passwords. A more secure way of managing local authentication is encrypting the passwords and storing them in a separate file.

In the case of a modern stand-alone Linux workstation, user information is stored in `/etc/passwd`, and password information is stored in `/etc/shadow`. Access control permissions allow anyone to read `/etc/passwd`, but only allow root to read the password hashes in `/etc/shadow`. Figures 4.2 and 4.3 contain a snippet of `/etc/passwd` and `/etc/shadow`, respectively.

Figure 4.2 Contents of `/etc/passwd` file

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
dallen:x:1000:1000:David Allen,,,:/home/dallen:/bin/bash
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
```

Figure 4.3 Contents of `/etc/shadow`

```
root:$1$6wDY0deM$teP1VcO3v9sCG5W4TRbLO.:12655:0:99999:7:::
daemon*:12590:0:99999:7:::
bin*:12590:0:99999:7:::
sys*:12590:0:99999:7:::
lp*:12590:0:99999:7:::
dallen:$1$hxo7GmPU$kt34Rhw2VV5cOdqjXb.qt.:12655:0:99999:7:::
nobody*:12590:0:99999:7:::
```

As you can see in Figure 4.3, the system-level accounts (such as *daemon*, *bin*, and *lp*) do not have a password. The password entries for *root* and *dallen* are hashes of the word “secret”.

Understanding NIS Authentication

Network Information Services takes the file-based approach of `/etc/passwd` and `/etc/shadow` a few steps further. In the case of NIS, a server stores network users, password, groups, and other information that corresponds to the information stored in local files.

However, NIS has a number of security disadvantages. For starters, there is no use of encryption. For this and other reasons, the use of NIS-based authentication is not recommended on networks. It was designed in the early days of network computing, and does not meet the needs of modern network security. Even Sun (the developer of NIS) has “end-of-lifed” NIS as of Solaris 10.

NIS+ is an improvement on NIS. While there is a NIS+ client for Linux, there is no open source NIS+ server for Linux, and Linux NIS+ development has stopped. Despite the shortcomings of NIS, it is still a widely used authentication mechanism, particularly on older UNIX networks that have not been updated to use more secure protocols.

Understanding Linux Client Authentication Settings

Up to this point, we have been discussing authentication capabilities of Linux, but have not delved into the configuration files on a Linux client. This section covers settings for two important files, `nsswitch.conf` and `ldap.conf`. In almost all versions of Linux, Solaris, BSD, MacOS, and HP-UX, these two files (along with PAM) control important authentication settings for LDAP authentication and glibc authentication.

Understanding Name Service Switch (`nsswitch.conf`)

Name Service Switch (NSS) is a C programming library developed by Sun that was designed to return the attributes of a user object. Later, NSS was extended to include the maps provided in Sun’s NIS service (`hosts`, `protocols`, `passwd`, `shadow`, `automount`, etc.).

Today, these are function calls in the GNU C Library (glibc). In Linux and other operating systems that use glibc, the configuration of NSS is controlled by the `nsswitch.conf` file located in the `/etc` directory. `Nsswitch.conf` contains a list of databases and their sources. When a glibc function is called to resolve these types of objects, each source is queried in the listed order. For the `nsswitch.conf`

file in Figure 4.4, the system is configured to first try local **files** for queries and then try **ldap** if local **files** cannot resolve the query.

Figure 4.4 NSS Configuration

```
#/etc/nsswitch.conf
passwd: files ldap
shadow: files ldap
group: files ldap
hosts: files dns
```

Additional types of databases, sources, and action items are available in `nsswitch.conf` to specify granular controls on the lookup process. Type **man nsswitch.conf** for more information.

Understanding LDAP Configuration Settings (`ldap.conf`)

LDAP configuration on a Linux system is primarily controlled by `/etc/ldap.conf`. This file contains the settings required to contact an LDAP server and manage authentication and query requests utilized by NSS. The settings in `ldap.conf` are used by the `nss_ldap` and `pam_ldap` modules.

Figure 4.5 shows the `ldap.conf` file present on the Fedora Linux desktops at Acme Widgets. In Fedora, these authentication file(s) can be easily changed using **authconfig**.

Figure 4.5 Acme Widgets Desktop `ldap.conf` File

```
#entry for server2 is in /etc/hosts file
host server2

#Acme Widgets base DN (directory suffix)
base dc=acmewidgets,dc=com

#Settings for nss_ldap
nss_base_passwd ou=Users,dc=acmewidgets,dc=com
nss_base_shadow ou=Users,dc=acmewidgets,dc=com
nss_base_group ou=Groups,dc=acmewidgets,dc=com
```

Understanding PAM

PAM is a flexible abstraction mechanism for user authentication that allows the authentication configuration for each application to be stored in a configuration file. This has the effect of separating the privilege-granting portions of an application into a system that can be dynamically reconfigured to support nearly any type of authentication scheme, without recompilation or reconfiguration of the application itself.

Invented by Sun Microsystems, PAM is used in the current version of Solaris. The Solaris PAM implementation differs slightly from the Linux PAM implementation in file location, error codes, and configuration methodology. In this book, when we refer to PAM, we are referring to Linux PAM. All modern Linux distributions, including Red Hat, Debian, Suse, Mandrake, and others, provide PAM support.

The PAM framework provides a library of functions for user authentication requests. When an application needs to authenticate a user, it calls the appropriate function, and the PAM subsystem handles the request. PAM will attempt to read the PAM configuration file for the application, which is usually stored as `/etc/pam.d/applicationname`. In some cases the configuration may be stored in a single file (**/etc/pam.conf**). If the `/etc/pam.d/` directory exists, most PAM implementations will ignore settings in `/etc/pam.conf`.

A PAM configuration file is composed of multiple lines listing PAM tokens formatted in this way:

```
service-name module-type control-flag module-path [args]
```

In systems with multiple PAM configuration files in `/etc/pam.d/application-name`, **service-name** is simply the name of the configuration file, and is not listed as an entry in the file. Configuration files of this type list the **module-type** token as the first item. Table 4.1 explains each of these token types and their usage.

Table 4.1 PAM Token Types and Descriptions

PAM Token	Description
service-name	The name of the service or application that is associated with this entry.
module-type	The type of PAM module (auth, account, session, or password).
control-flag	Controls how the module will react to the success or failure of a PAM request.
module-path	The relative or absolute path to the module.
args	The arguments to be passed to the module (not all modules use arguments).

PAM configuration files separate the authentication functions into four types: user authentication (*auth*), account restriction enforcement (*account*), session startup/teardown (*session*), and password (or authentication token) updating (*password*) functions. The characteristics of these four PAM module types are explained below in Table 4.2.

Table 4.2 PAM Module Types and Descriptions

PAM Module Type	Description
auth	Provides authentication services to establish identity and grant privileges. An auth module verifies identity using credentials such as a password, biometric, hardware token, or other authentication mechanism.
account	Manages the non-authentication aspects of the account. An account module may restrict or deny access based on password expiration, time of day, or system resource availability.
session	Manages aspects of a request that need to be performed prior to a user being granted access to a service, or after the user has finished using a service. A session module typically performs tasks such as audit logging or mounting/unmounting a home directory.
password	Used for updating the authentication token associated with the user. A password module performs the function of changing a user password.

PAM control flags define how the success or failure of a PAM module will affect the PAM stack. Table 4.3 lists the four types of control flags with a description of each flag.

Table 4.3 PAM Control Flags and Descriptions

PAM Control Flag	Description
requisite	Failure of a requisite module immediately terminates the authentication process and results in authentication failure.
required	Failure of a required module will lead to authentication failure, although the rest of the stack will be executed.
sufficient	Success of a sufficient module will lead to authentication success, even if a previous required module has failed. Failure of a sufficient module does not lead to authentication failure.
optional	Success or failure of an optional module will not affect authentication success or failure, unless the optional module is the only listed module.

For further information about PAM control flags, modules, and arguments, type **man pam** or **man pam.conf**, or visit www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/ref-guide/s1-pam-sample-simple.html.

Designing Linux-Based Authentication Services

Designing Linux-based authentication services is often a relatively straightforward process. Usually, it is simply a matter of determining the authentication requirements for the organization's computer systems and deploying the appropriate servers, as well as determining the sizing, number, and placement of authentication servers.

While design of authentication services is often straightforward, the importance of “getting it right” is significant. Authentication performs a critical security role – figuring out if a user is who s/he claims to be. In almost all cases this is done with a username–password combination. Although this style of authentication is well understood and ubiquitous, there are some security considerations, particularly around management of passwords.

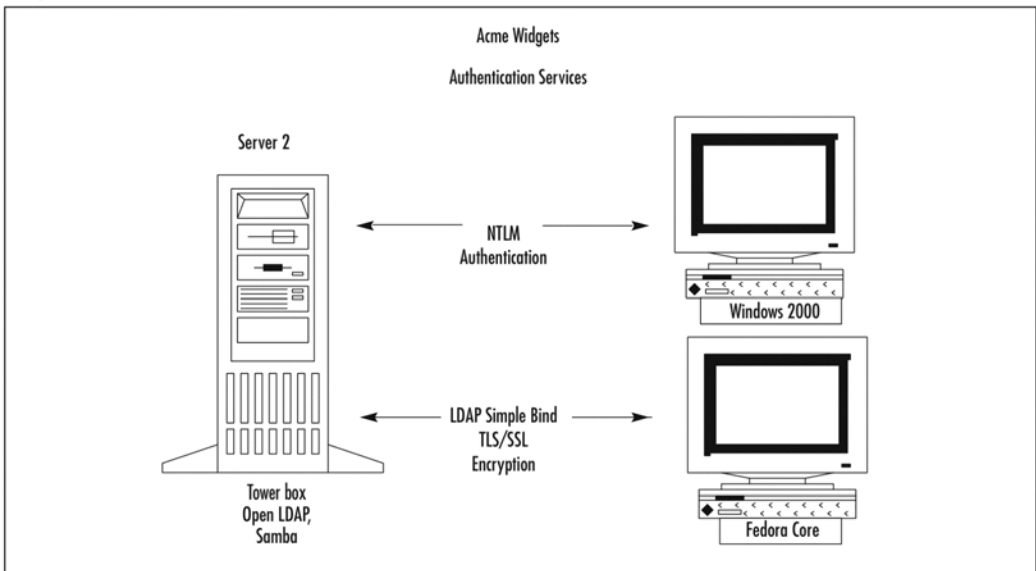
If there is one service where mistakes can be costly, authentication is most assuredly that service. If authentication services have failed in an organization, in many cases systems will not accept new logons, and existing sessions will begin to expire. If the authentication service stores credentials in a directory, and the directory server goes offline or the network connection between the authentication server and the directory server (if running on different machines) is unavailable, in most cases the authentication service will cease to function.

These potential problems need to be addressed when designing an authentication infrastructure. If the impact to a business is costly when authentication services are not functioning (which it is in most businesses with more than a handful of employees), the appropriate design guideline is to increase the availability of the authentication services by building redundancy into the infrastructure. In addition to using fault-tolerant hardware, the use of more than one authentication server (and multiple directory servers) is strongly recommended for most businesses, and required for large organizations.

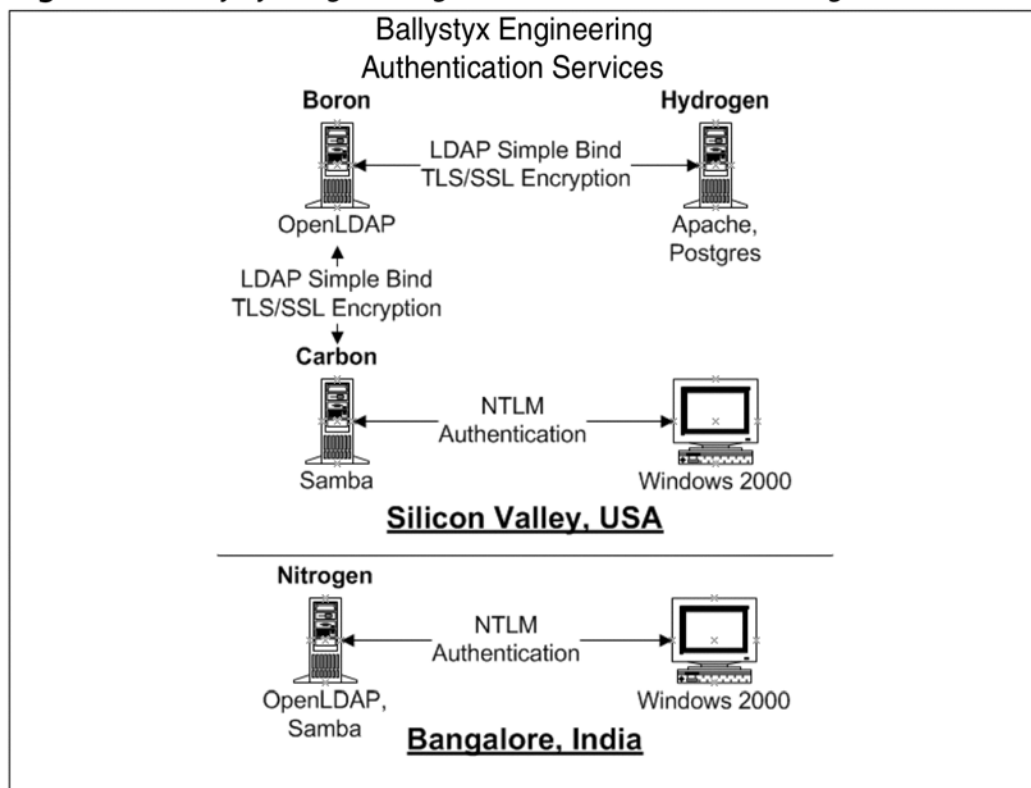
To ensure better security, the mandatory requirement of strong passwords and mandatory user password changes is recommended. While this does make passwords slightly more difficult to remember, it provides additional layers of defense against a malicious intruder or misconfigured server. Even if an intruder manages to obtain the encrypted passwords, s/he will not be able to decrypt them before the next required password changes. All security-conscious organizations should require users to change their passwords at regular intervals. The interval varies based upon the industry and other business specifics, but for most corporations the password change interval will be somewhere between one and six months.

Designing Cross-Platform Authentication Services

Authentication services design specifics will vary depending on the requirements. The text and figures below describe the authentication solutions for Acme Widgets and Ballystyx Engineering. Because Acme Widgets needs to authenticate Windows and Linux desktops, the authentication services must work seamlessly for both platforms. For this reason Acme Widgets is using a Linux-based authentication solution that utilizes OpenLDAP to store authentication credentials and provide LDAP authentication, and Samba to provide NTLM authentication services to Windows clients. Figure 4.6 illustrates the design of the authentication services for Acme Widgets.

Figure 4.6 Acme Widgets Authentication Services Design

The authentication services design for Ballystyx Engineering follows the same principles as Acme Widgets' authentication services design, but is scaled to meet the needs of a global company. Like Acme Widgets, Windows 2000 desktops will utilize NTLM authentication, and Linux systems will authenticate using LDAP simple binds with TLS encryption. Figure 4.7 illustrates the authentication services design for Ballystyx Engineering.

Figure 4.7 Ballystyx Engineering Authentication Services Design

As budget becomes available, Vijay intends to deploy additional servers to ensure high-availability for authentication services. Redundant servers will ensure that authentication is always available, which will ensure that Vijay will continue to enjoy a restful sleep if one of the servers fails in the middle of the night.

Installing and Configuring Samba

Installation and configuration of Samba is the first step to provision NTLM authentication. On a test lab server, compile and install the current Samba server package (3.0.6 as of this writing) or install the current Samba server package from your Linux distribution. If you are considering compiling the latest Samba package yourself, keep in mind that using the Samba package provided by your Linux distribution will be much easier and generally will “just work” as long as the appropriate features are provided.

After you have installed Samba, start your favorite text editor and edit `smb.conf`. Optionally, you may use a graphical configuration tool like Webmin (www.webmin.com) or SWAT (Samba Web Administration Tool) to configure `smb.conf`. Figure 4.8 shows the `smb.conf` file for Acme Widgets. We recommend using the sample provided in the Windows to Linux Migration Toolkit as a starting point.

Figure 4.8 Acme Widgets `smb.conf` file

```
[global]
    workgroup = ACMEWIDGETS
    netbios name = SERVER2
    server string = Samba Server
    passdb backend = ldapsam:ldap://127.0.0.1/
    idmap backend = ldap:ldap://127.0.0.1/
    ldap admin dn = cn=Manager,dc=acmewidgets,dc=com
    ldap suffix = dc=acmewidgets,dc=com
    ldap group suffix = ou=Groups
    ldap user suffix = ou=Users
    ldap machine suffix = ou=Computers
    ldap idmap suffix = ou=Users
    ldap ssl = no
    security = user
    domain logons = yes
    domain master = no
    log file = /var/log/samba/%m.log
    max log size = 50
    socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192
    dns proxy = No
    wins support = No
    preferred master = Yes
    add user script = /usr/local/sbin/smbldap-useradd -m "%u"
    ldap delete dn = Yes
    delete user script = /usr/local/sbin/smbldap-userdel "%u"
    add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
    add group script = /usr/local/sbin/smbldap-groupadd -p "%g"
    delete group script = /usr/local/sbin/smbldap-groupdel "%g"
```

```

add user to group script = /usr/local/sbin/smbldap-groupmod -m "%u" "%g"
delete user from group script = /usr/local/sbin/smbldap-groupmod -x "%u"
"%g"
set primary group script = /usr/local/sbin/smbldap-usermod -g "%g" "%u"

```

Acme Widgets uses the same host for Samba services as well as LDAP services. If these applications are installed on multiple machines, the configuration file needs to reference the other server by DNS (Domain Name System) name or IP (Internet Protocol) address. These settings for `smb.conf` can be fully explained by typing **man smb.conf**, but some of the settings are important enough to explore in this section. In the first few lines we can see that this installation of Samba is LDAP-enabled, and the settings for the LDAP portions of the `smb.conf` file correspond to the directory server `slapd.conf`. The LDAP directory admin dn (rootdn in `slapd.conf` parlance) is listed, as well as the suffix, and the organizational units that contain users, groups, and computers. These settings correspond to the DIT (Directory Information Tree) Acme Widgets set up in Chapter 3. The **domain logins = yes** and **domain master = no** settings also merit explanation. This configuration sets up the Samba server to be a backup domain controller in order to be authorized to receive a copy of the authentication objects from the existing Windows domain. When the migration is complete and the Windows PDC is retired, Sam will change the **domain master** entry to **yes** and restart Samba. This will transform the Samba server from a BDC to a PDC. This procedure is detailed in the migration section below.

The script settings in `smb.conf` correspond to the IDEALX `smbldap` tools that will be installed in the migration section of this chapter. Please note that the physical path to the `smbldap` toolkit may vary depending on how the toolkit was installed. If the toolkit was installed from the `tar.gz` file, the path will be `/usr/local/sbin`. If the toolkit was installed from an RPM package, the path will be `/usr/sbin`. The easiest way to determine this information is to type **which smbldap-useradd** at a command prompt and note the path. If these settings are not correct, your Samba migration and ongoing maintenance will not function properly!

Figure 4.9 shows the Samba network file share listings in `smb.conf`, corresponding to the home directory, network logon, and roaming profile storage for Windows logon services.

Figure 4.9 Acme Widgets Samba Shares

```
[homes]
    comment = Home Directories
    read only = No
    browseable = No
    create mask = 0644
    directory mask = 0775

[netlogon]
    path = /home/netlogon/
    browseable = No
    read only = yes

[profiles]
    path = /home/profiles
    read only = no
    create mask = 0600
    directory mask = 0700
    browseable = No
    guest ok = Yes
    profile acls = yes
    csc policy = disable
    # secures profiles by user
    force user = %U
    # allow administrative access to profiles
    valid users = %U @"Domain Admins"
```

After you have set up the Samba server, ensure that it starts up properly. For most Linux distributions, the following commands will start Samba services and show status details:

```
/etc/init.d/smb start
/etc/init.d/smb status
/usr/bin/smbstatus -v
```

Assuming there are no errors, your Samba configuration should be correct. In order to run the migration scripts, Samba services must be turned off. Before

proceeding to the migration section, stop your Samba server. In most Linux distributions, the follow command will stop Samba services.

```
/etc/init.d/smb stop
```

Migrating from NT/Exchange or Active Directory

This section details the directory and authentication services migration from Windows NT, Exchange 5.5, Windows 2000, and/or Active Directory to Linux-based directory and authentication services. This section requires that you have a properly configured Samba and OpenLDAP server(s). Samba must be shut down, but OpenLDAP must be running.

The process of migrating a Windows NT domain (with or without Exchange) and Active Directory are functionally identical. This process is performed in two steps. In the first step, the scripts obtain a copy of the authentication information and insert these objects into the directory. In the second step, the scripts obtain items such as contacts and distribution groups, as well as non-authentication user information. This information is then inserted into the directory as appropriate.

Preparing for Migration

You will need to perform a few more steps to prepare your server(s) for migration. First, install the `smldap-tools` from IDEALX at <http://samba.idealx.org>. Depending on which package is installed, this will populate `/usr/local/sbin/` or `/usr/sbin` with the `smldap` scripts, and `/etc/smbldap-tools/` with the configuration files. Make sure you use at least version 0.8.5 of `smldap-tools`. Although the experience is educational, configuring `smldap-tools` is not necessary at this time because the migration scripts below will do this for you.

Because we will be using Samba components to help obtain the authentication objects, and because this server will likely be used to maintain samba accounts in the future, it is very important to make sure that the server has been properly configured to authenticate to the target LDAP server. This is typically accomplished via the `/etc/pam.d/samba`, `/etc/ldap.conf`, and `/etc/nsswitch.conf` files, described above in the “Understanding Linux Authentication” section. To achieve this on Acme Widgets’ Fedora desktops, Sam (as root) simply typed **authconfig** and entered the appropriate LDAP values.

NOTE

Failure to properly configure the authentication components mentioned above will result in the migration scripts populating the directory with unusable posix account information, and will likely break the Samba server's authentication configuration.

At this point in the migration process, you should have a properly installed and configured Samba server (*not running*) and the IDEALX smbldap toolkit installed (unconfigured is fine) on the local machine. Additionally, the OpenLDAP server should have a properly configured slapd.conf file as outlined in Chapter 3, and *the directory server SHOULD be running*. For a small company like Acme Widgets, all of these services are running on the same server. At Ballystyx Engineering, OpenLDAP and Samba run on multiple servers.

You can now skip to the appropriate section below depending on whether you are migrating from Windows NT or Windows 2000.

NT / Exchange 5.5 Migration Path

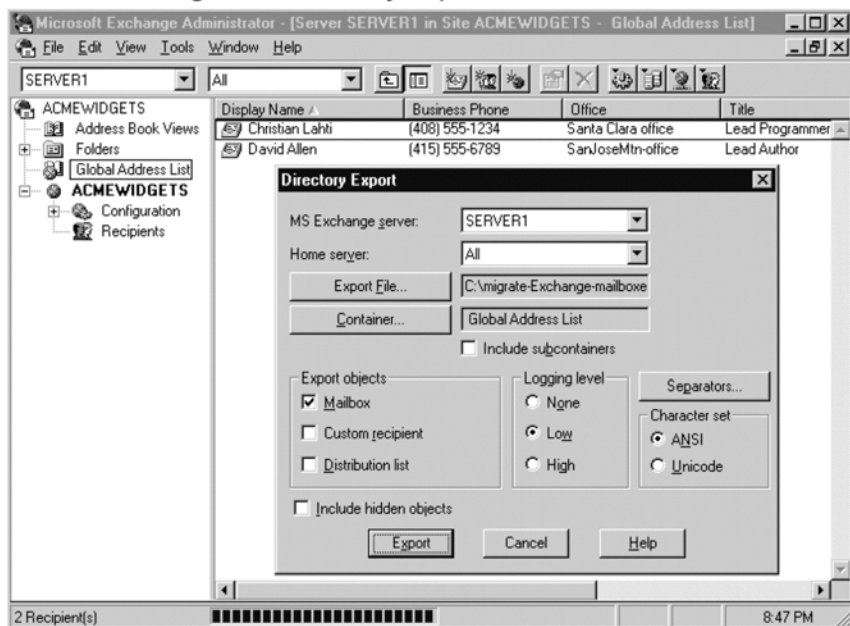
In Windows NT, the SAM database stores only a limited amount of user information. Information about a user's e-mail, location, telephone, and other contact information must be stored external to the SAM, in the Exchange 5.5 directory. To migrate Exchange 5.5 information, the scripts will need a *user to mailbox map* to match authentication objects with the corresponding Exchange information. This map may be obtained through the Directory Export feature of the Exchange 5.5 Administrator program. The use of an external file is necessary because the mapping between an Exchange mailbox and the associated Windows NT account is not exposed via the Exchange 5.5 LDAP interface. If you are not migrating from Exchange and/or only wish to populate the authentication objects, you can safely skip the next step.

To extract information from the Exchange 5.5 directory, launch the Exchange 5.5 Administrator program (typically c:\exchsrvr\admin.exe). To start the Administrator program, choose **Start | Programs | Microsoft Exchange | Microsoft Exchange Administrator**.

In order to extract the user mailbox information from the Exchange directory, perform a directory export of all of the user mailboxes in the Global Address List (GAL) by choosing **Tools | Directory Export**. Figure 4.10

illustrates the Exchange 5.5 administrator program preparing to do a directory export of Acme Widgets.

Figure 4.10 Exchange 5.5 Directory Export



Note that only mailbox objects have to be exported, since this provides the necessary account map information needed by the migration process. The remainder of the group, contact, and user information will be obtained via LDAP.

You are now ready to run the first script, `w2lmt-migrate-smbauth`, which is designed to extract authentication information. This script requires the use of a configuration file, although future versions of the script may prompt for information if no configuration file is given. Figure 4.11 lists the configuration file for Acme Widgets.

Figure 4.11 Acme Widgets migrate-smbauth Configuration File

```
#migrate-smbauth-acmewidgets.conf
#information about source server and target server
SourceHost="server1"
SourceDomain="ACMEWIDGETS"
SourceType="NT4"
```

```
SourceAdminAccount="Administrator"
TargetHost="localhost"
TargetPort="389"

#locations of config files for samba and smbldap-toolkit
smbconf="/etc/samba/smb.conf"
smbldap="/etc/smbldap-tools/smbldap.conf"
```



The information in the migrate-smbauth configuration is relatively straightforward. The script has to know where to find the configuration files to read/modify, and which hosts are involved with the migration. Sam uses the command below to perform the migration on Acme Widgets' servers.

```
/<path_to_scripts>/w2lmt-migrate-smbauth -f migrate-smbauth-acmewidgets.conf
```

This command performs the following functions:

- Check to make sure it has everything it needs to continue.
- Prompt for passwords for both the source NT and target OpenLDAP servers.
- Enumerate the domain SID.
- Prompt for default settings (with a change/save option).
- Join the NT domain as a BDC.
- Create top-level OUs in OpenLDAP if needed.
- Invoke Samba components to extract users, groups and machine objects and insert them into the OpenLDAP server.
- Process group membership information and modify group objects to add uid membership information.



With the directory populated with user auth objects, the next step is to run `w2lmt-directory-auth`. This script will query the Exchange server in order to update the directory objects with any pertinent information stored there. The script also adds contact and distribution group objects. If you are not running Exchange or do not wish to migrate this information to your new Linux environment, you may skip this step. This script also requires a short configuration file. Figure 4.12 shows the migrate-directory configuration file for Acme Widgets.

 **Figure 4.12** Acme Widgets migrate-directory Configuration File

```
#migrate-directory-acmewidgets.conf


#information about target LDAP instance
SlapdPath="/etc/openldap/slapd.conf"
TargetPort="389"

#specify the relative DN here for user objects in the target LDAP instance.
# you can use a different DN if you want your exchange contacts separate
from
# your User/Group DNs. (ex.) Users would equal
'ou=Users,dc=acmewidgets,dc=com'
UsersOU="ou=Users"
ContactsOU="ou=Contacts"
GroupsOU="ou=Groups"
DistributionGroupsOU="ou=DistributionGroups"

#information about import directory (if any)
#valid entries are EXCH55, AD
SourceType="EXCH55"
SourceHost="server1"
SourceAdminAccount="Administrator"

#if import directory is exchange55, a map file is required
Exchange55CSV="/tmp/acmewidgets-exch55.csv"
```

The configuration information above shows the Exchange host that the script will query, and describes where (in the DIT) to place contacts and distribution groups. This script gets some of its information from the slapd.conf file, so it has to be run on the directory server. Here is how Sam invoked the script at Acme Widgets:



```
/<path_to_scripts>/w2lmt-migrate-directory -f migrate-directory-
acmewidgets.conf
```

The script performs the following actions:

- Check to make sure it has everything it needs to continue.



- Prompt for passwords for both the source Exchange and target OpenLDAP servers.
- Add any necessary top-level OUs for contacts and distribution groups to OpenLDAP.
- Read in the provided Exchange export (.csv) file.
- Enumerate all of the person mailbox objects.
- Determine whether the object is a contact or mapped NT uid (note, with multiple mailboxes that match the same uid, the last one wins and this mailbox information is what will remain for that uid).
- Either update matching user object in OpenLDAP with attributes such as address, mail, phone, etc. OR add this as a contact if applicable.
- Enumerate distribution groups.
- Create objects in OpenLDAP populated with groupOfNames attributes.
- Process distribution group membership information with either contacts or user DNs as applicable.

You should now have a fully populated directory containing all of the authentication and directory information from your previous NT domain and Exchange 5.5 instance. The only thing left to do is to change your samba smb.conf setting to make your Linux server the PDC for the domain, turn off the original PDC, and restart your samba services.

Active Directory Migration Path

You are now ready to run the first script, w2lmt-migrate-smbauth, which is designed to extract authentication information. This script requires the use of a configuration file, although future versions of the script may prompt for information if no configuration file is given. Figure 4.13 lists the configuration file for Ballystyx.



NOTE

If you are running Active Directory in *Mixed Mode*, you should use the NT4 setting as the **SourceType** in the w2lmt-migrate-smbauth script configuration file. By using this setting, you will preserve the SIDs and RIDs of your existing domain. If you are running Active Directory in

Native Mode, the scripts will create a new domain with new SIDs and RIDs for users, groups, and machines. This is because Active Directory Native Mode is not compatible with the Samba NT4-style domain that is being migrated to. Keep in mind that all machines joined to this domain must be RE-joined to the new Linux-based domain.



Figure 4.13 Ballystyx migrate-smbauth Configuration File

```
#migrate-smbauth-ballystyx.conf

#information about source server and target server
SourceHost="beryllium"
SourceDomain="BALLYSTYX"
SourceType="AD"
SourceAdminAccount="Administrator"
TargetHost="localhost"
TargetPort="389"

#locations of config files for samba and smbldap-toolkit
smbconf="/etc/samba/smb.conf"
smbldap="/etc/smbldap-tools/smbldap.conf"
```

The above information is fairly straightforward; the script has to be told where to find the configuration files to set up and the hosts it will be interfacing with for the migration. The command below shows how Vijay invokes the script on the Samba server at Ballystyx Engineering:



```
<path_to_scripts>/w2lmt-migrate-smbauth -f migrate-smbauth-ballystyx.conf
```

The script will perform the following functions:



- Check to make sure it has everything it needs to continue.
- Prompt for passwords for both the source NT and target OpenLDAP servers.
- Enumerate the domain SID.
- Prompt for default settings (you have a chance to change/save them).
- Join the NT domain as a BDC.



- Create top-level OUs in OpenLDAP if needed.
- Enumerate group objects.
- Create group objects in OpenLDAP populated with `posixGroup` and `sambaGroupMapping` attributes.
- Enumerate machine account objects (hosts joined to existing domain).
- Create a machine (account) object in OpenLDAP with populated attributes for `inetOrgPerson`, `posixAccount`, and `sambaAccount`.
- Enumerate all of the authentication objects (users).
- Create user objects in OpenLDAP with populated attributes for `inetOrgPerson`, `posixAccount`, and `sambaAccount`.
- Process group membership information and modify group objects to add uid membership information.

The next step is to query the Active Directory server in order to update the above objects with any non-authentication-related information, such as e-mail, phone numbers, and other contact information, as well as to obtain contact object and distribution group information. If you do not care about retaining this information in your new Linux environment you can skip this step. The next script also requires a short configuration script. Figure 4.14 lists the migrate-directory configuration file for Ballystyx.



Figure 4.14 Ballystyx migrate-directory Configuration File

```
#migrate-directory-ballystyx.conf
#information about target LDAP instance
SlapdPath="/etc/openldap/slapd.conf"
TargetPort="389"


#specify the relative DN here for user objects in the target LDAP instance.
# you can use a different DN if you want your exchange contacts separate
from
# your User/Group DNs. (ex.) Users would equal
'ou=Users,dc=ballystyx,dc=local'
UsersOU="ou=Users"
ContactsOU="ou=Contacts"
GroupsOU="ou=Groups"
```

```
DistributionGroupsOU="ou=DistributionGroups"

#information about import directory (if any)
#valid entries are EXCH55, AD
SourceType="AD"
SourceHost="beryllium"
SourceAdminAccount="Administrator"


#if import directory is exchange55, a map file is required
Exchange55CSV=
```

The above information basically provides the script with the Active Directory host it will be querying, as well as where (in the DIT) to place contacts and distribution groups. This script gets some of its information from the `slapd.conf` file, so it has to be run on the directory server. The command below shows how Vijay invokes the script on the Samba server at Ballystyx Engineering:



```
/ <path_to_scripts>/w2lmt-migrate-directory -f migrate-directory-  
ballystyx.conf
```

The script will perform the following functions:

- 
- Check to make sure it has everything it needs to continue.
 - Prompt for passwords for both the source Exchange and target LDAP servers.
 - Add any necessary top-level OUs for contacts and distribution groups to LDAP.
 - Enumerate distribution groups.
 - Create objects in OpenLDAP populated with `groupOfNames` attributes.
 - Enumerate all of the person mailbox objects.
 - Update matching user object in OpenLDAP with attributes such as address, mail, phone, etc.
 - Process distribution group membership information for user.
 - Enumerate all of the contact objects.

- Create contact object in OpenLDAP with attributes for inetorgPerson, mail, phones, etc.
- Process Distribution Group membership information for contact.

You should now have a fully populated directory containing all of the authentication and directory information from your previous NT domain and Exchange 5.5 instance. The only thing left to do is to change your Samba `smb.conf` settings to make your Linux server the PDC for the domain, turn off the original PDC, and restart your samba services. Since this is a new domain, you will need to re-join your remaining Windows 200x servers and workstations that will survive the migration (if any!) to the new domain.

Migrating Windows Logon Authentication Files

Any logon scripts that are being used by the organization must be copied from the NETLOGON share (usually found in `C:\WINNT\System32\Repl\Import\Scripts\`) to the new location of the NETLOGON share on the Linux Samba server (the default is `/home/netlogon/`).

In addition, if roaming profiles are being used, the files and directories found in the users' profiles (including `NTUSER.DAT` or `USER.DAT`, `NTUSER.INI`, and others) directory must also be copied to the appropriate shared directory on the Linux Samba server (default is `/home/profiles/`).

Home directories are not strictly used for Windows logon, although Windows provides a drive mapping for the user home directory. Migration of the contents of user home directories is best handled in Chapter 5, "File Services," when migrating the contents of other files and directories.

Testing Authentication Services

After all of this migration work is complete, the next step is testing your Linux-based authentication services. Try to login from a workstation using the new Samba server you've set up (in the lab) to authenticate. If this test works, try logging in using other usernames, and make sure that logon scripts are being executed, and user (roaming) profile information is being properly updated.

Enabling Encryption

Prior to using these systems for authentication in an enterprise setting, you must enable encryption. Because enabling encryption is a complex task, we recommend that you get authentication services working properly without encryption

initially. Test logon and other authentication features using Windows and Linux computers. After you have verified that authentication services are working properly, you are ready to add encryption capabilities to ensure appropriate security.

The steps involved with enabling encryption primarily consist of the following

1. Obtain and install a valid server certificate.
2. Configure slapd.conf to use the server certificate and require TLS.
3. Configure smb.conf to utilize TLS.
4. Configure clients to utilize TLS.
5. Test, test, test!!

The server certificate that you use can be obtained from any Certificate Signing Authority (CSA) such as Verisign or CA-Cert, or you can generate your own via OpenSSL. The OpenSSL certificate creation process is covered in Chapter 9, “Web Services”. After you have installed a server certificate, you need to modify the server configuration file to utilize the new certificate. The additional configuration information required to enable TLS for Acme Widgets is listed in Figures 4.15, 4.16, and 4.17.

Figure 4.15 Acme Widgets slapd.conf Encryption Configuration Section

```

TLSCertificateFile /etc/openldap/server.pem
TLSCertificateKeyFile /etc/openldap/server.key
TLSCACertificateFile /etc/openldap/ca.pem
TLSCipherSuite :SSLv3
TLSVerifyClient demand

```

Figure 4.16 Acme Widgets smb.conf Encryption Configuration Directive

```

ldap ssl = yes

```

Figure 4.17 Acme Widgets Desktop Client ldap.conf

```

ssl                start_tls
TLS_REQCERT        allow

```

If configuration files contain `ldap://` URI references, they must be changed to `ldaps://` URIs. Make sure that the proper TLS options are present when `slapd` is restarted. For example,

```
slapd -h ldap://127.0.0.1/ ldaps:///
```

generally works well. Fedora users would update the `/etc/sysconfig/ldap` file with the `ldaps:///` option.

The next step is to retest authentication services with encryption enabled. A quick test at Acme Widgets to make sure that TLS is working is to type:

```
ldapsearch -x -H ldaps://localhost -b 'dc=acmewidgets,dc=com'  
'(objectclass=*)'
```

Summary

Authentication is one of the most important network services. Being able to determine the identity of a user and control access to network resources forms a core security underpinning of a corporate network. In most cases, authentication of users consists of validating credentials consisting of one or more factors: something you know (username, password), something you have (token), or something you are (biometric). With an OpenLDAP directory serving as the centralized repository of authentication information, you can build flexible authentication methodologies that allow for centralized control of distributed heterogeneous authentication services.

Microsoft utilized LANMAN protocols for authentication in Windows 98 and NT, and added Kerberos authentication services for Windows 2000/XP. There are currently no open source authentication servers that provide Kerberos logon services to out-of-the-box Windows 2000/XP clients. Samba servers provide NTLM / LANMAN authentication services that may be used by the entire family of Windows clients and servers (Windows 98/NT/2000/XP).

Migration of Windows NT and Exchange 5.5 information is usually straightforward, especially when there is a direct correspondence between NT accounts and Exchange mailboxes. The migration scripts will easily knit these two directories together to present a unified view in OpenLDAP of all the user data. The new Samba-managed domain provides all of the functionality of a Windows NT domain, and more.

Migration of Windows 2000 Active Directory information generally works well, but if you are using Exchange 2000 or were unable to migrate your AD-integrated MS-DNS services in the previous chapter, you must run your Linux-based directory services in parallel until you can migrate the Active Directory dependencies. If you are using Active Directory in *Native Mode*, you will need to re-join all of the Windows computers to the new Linux-managed domain.

Solutions Fast Track

Understanding Windows Authentication

- ☑ To logon to a Windows computer, the user inputs a username, password, and the domain. Windows 98 authenticates using LANMAN, Windows NT authenticates using NTLM, and Windows 2000/XP will attempt

Kerberos authentication with a backwards-compatible fallback to NTLM.

- ☑ Following successful authentication, the Windows computer will update the roaming profile (if configured) and run logon scripts.
- ☑ Windows 2000 may use Group Policy Objects to apply configuration settings and launch (or install) applications.

Understanding Linux Authentication

- ☑ LDAP authentication is one of the most versatile, well-supported, and widely used cross-platform enterprise authentication technologies. The utility of LDAP authentication can be extended using Simple Authentication and Security Layer (SASL).
- ☑ One of the simplest ways to authenticate is using local files. Using `/etc/passwd` to store user information and `/etc/shadow` to store user passwords is the most common way to achieve local authentication.
- ☑ `/etc/nsswitch.conf`, `/etc/ldap.conf`, and `/etc/pam.d/applicationname` contain configuration information for authentication mechanisms used on Linux (and many *nix) computers.
- ☑ Pluggable Authentication Modules, or PAM, is a flexible abstraction method for user authentication. PAM separates the privilege-granting portions of an application into a system that can be dynamically reconfigured by modifying `/etc/pam.d/applicationname`.

Designing Linux-Based Authentication Services

- ☑ The first step in designing Linux-based authentication services is determining the authentication requirements for the organization's computer systems. For most organizations, this means authenticating using LDAP/TLS for Linux systems, and NTLM (provided via Samba) for Windows systems.
- ☑ The next step is to determine the type, placement, and number of authentication servers in the organization. For a small company like Acme Widgets, one authentication server is sufficient. For a larger company like Ballystyx Engineering, at least one authentication server

per site is required. For fault-tolerant installations, at least two servers per site are required.

- ☑ The final step prior to deployment is to install, configure, and test authentication services in the lab. Install Samba and configure `smb.conf`.

Migrating Information from NT/Exchange 5.5/AD

- ☑ To prepare for migration, ensure that the Samba server is configured to authenticate using LDAP, and install the `smbldap-tools` from IDEALX at <http://samba.idealx.org>.
- ☑ If you would like to use information from Exchange 5.5 for migration, perform a directory export of the mailboxes in the Exchange 5.5 Global Address List (GAL) by choosing **Tools | Directory Export** from Microsoft Exchange Administrator.
- ☑ To populate the Samba and OpenLDAP with user, group, and computer information migrated from NT / Exchange or Active Directory, run the `migrate-smbauth` and `migrate-directory` scripts. Be sure to follow the instructions carefully, and check <http://sourceforge.net/projects/w2lmt> for the latest versions of the scripts.
- ☑ After verifying the functionality of authentication services by testing in the lab, enable encryption to provide additional security. Obtain and install an OpenSSL server certificate, and configure applications to enable TLS encryption.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: My system has TWO ldap.conf files. Why?

A: `etc/ldap.conf` is used by `pam_ldap` and `nss_ldap` for system-wide configuration of LDAP parameters, including authentication. `/etc/openldap/ldap.conf` (or `/etc/ldap/ldap.conf`) is used by OpenLDAP utilities such as `ldapsearch`.

Q: Why must I re-join my Windows 200x to the Samba-managed domain after I migrate from Active Directory running in Native Mode?

A: Because Native Mode Active Directory does not use the same SID/RID scheme used by the NT SAM, it is not possible to migrate the object descriptor IDs into the Samba-managed NT-style domain.

File Services

Topics in this Chapter:

- Understanding Windows File Systems
- Understanding Linux File Systems
- Understanding Permissions Management (Access Control)
- Understanding File Backup, Restore, and Replication Options
- AMANDA
- Designing Linux-based File Services
- Migrating File Services to Linux
- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

File services is a term that is used when talking about accessing files on a remote system. Regardless of computer platform, most companies use a networked repository for commonly accessed data, including home directories, department files, and company resource files. Access to networked data must be protected from unauthorized access using access control, must be protected from loss or damage with regular backups, must be served with adequate performance for end user usage requirements, and must be organized in a manageable way that can handle growth. While many types of file services are available in Linux, Samba is the most tangible solution for supporting Windows clients with a Linux server. The challenge for this conversion is to have a smooth migration of the data, shares, and access control lists (ACLs), while maintaining performance and ensuring data security.

Understanding Windows File Systems

File services offer network access to data stored on file systems. Each type of file system has features and limitations that affect the way file services can share the data. The Modern Microsoft Operating System (OS) offers local compatibility with FAT16, FAT32, NTFS4, and NTFS5 file systems, and the ability to share files on these file systems with the network using a protocol called Common Internet File Services (CIFS). This section discusses the details regarding each of these file systems, pointing out the benefits and differences of each, and also covers the history behind the file system evolution Microsoft has followed.

Why do we need file systems? Disk drive surfaces are divided into address locations (sectors). Partitions can be created on the disk by grouping a number of sectors. When a partition is formatted with a file system, the file system attempts to organize how the data will be organized, written, found, accessed, and checked within the sectors of the partition.

Understanding Windows File Allocation Table File Systems

A File Allocation Table (FAT)-formatted partition groups the sectors into clusters and each file occupies one cluster. All FAT file systems have a maximum limit of clusters that can be in a partition. The cluster limit is based on the number of bits used to address sections of the cluster minus a few reserved bits. As hard drives

increase in size there are more sectors on the disk. To create a larger partition (across an entire disk), a cluster must contain more sectors. A lot of space is wasted if a file is less than the size of a cluster (sectors on the disk remain unusable).

Tools and Traps

FAT Bits

- FAT12 uses 12 bits, or 2^{12} bits or about 4,096 clusters.
- FAT16 uses 16 bits, or 2^{16} bits or about 65,536 clusters.
- FAT32 uses 28 bits, or 2^{28} bits or about 268 million clusters.

Additionally, there is a limit on the size of each cluster. In order to use an entire disk on a large hard drive, you must create many partitions and remember which partition contains the stored data.

Tools and Traps

FAT Partitions

- FAT12 can create 16MB partitions.
- FAT16 can create 2GB partitions in DOS 4.0 and up, Windows 9x, and ME, or 4GB partitions with NT.
- FAT32 can create 8 terabytes (TB) partitions, but only a 32GB partition is possible with versions of Windows up to XP.

Tools and Traps

Maximum Files in a Directory

- FAT - 512 files or folders per folder
- FAT32 - 65,534 files or folders per folder
- NT File System (NTFS) - 4,294,967,295 files or folders per folder

By default, FAT stores a backup copy of the FAT table in the partition header; with Fat32, this can be disabled to speed up access. All FAT versions lack the concept of user ownership and user level access control, file locking, and redundancy and can fragment easily. FAT stores some basic file attributes such as *read only*, *hidden*, *system file*, *volume label*, *subdirectory*, and *archive*. There are three naming rules for files stored on a Fat12 or Fat16 partition.

1. The name is limited to an 8-character prefix, followed by a period and a 3-character suffix (i.e., *my_files.txt*).
2. The name must begin with a letter or number.
3. The case will not be preserved.

VFAT/FAT32 has the same 2.0GB maximum file system size limit that FAT16 has and adds three improvements:

1. Long file name support (255 characters long including spaces and multiple periods; preserves the case but is not case-sensitive).
2. Improved performance using “32-bit protected mode access.”
3. Better management capabilities; disk locking for “exclusive mode” disk access by a program to a file.

Virtual File Allocation Table (VFAT)/FAT32 offers the following advances:

- Uses 32 bits per FAT entry and a smaller cluster size to increase the maximum file system size to 32GB.
- Each individual file occupies a single cluster; a small file with a smaller cluster size wastes less space.

- An extension to the VFAT that shares the VFAT file name limitation.
- Utilities were made available to convert VFAT to FAT32 in a one-way operation.
- FAT32 partitions are larger than 32GB, are prepared by other OS', and are usable by windows.

Windows NT can use a 64 Kbits cluster and a 64 Kbits cluster, extending the maximum file system size to 4GB for a VFAT partition.

With an FAT file system, network file-sharing ACLs are limited to *full control*, *change*, and *read*.

In 1988, FAT16 became available, and is used with DOS 4.0 and up, Windows 3.x, Windows 95 OEM SR1, Windows ME, and Windows NT. By 1995, VFAT became available, and can be used with Windows 3.x, Windows 95 OEM SR1, Windows ME, and Windows NT. Around 1998, FAT32 became available, and can be used with Windows 95 OEM SR2, Windows 98, Windows ME, Windows 2000, and Windows XP.

Understanding Windows NTFS File Systems

With Windows NT4, Microsoft included a more complex file system called NTFS. NTFS uses a Master File Table (MFT) to track each file in the partition, with a security descriptor (security and permissions) for each file. Each security descriptor contains a System Access Control List (SACL) for auditing and a Discretionary Access Control List (DACL) affecting access to the file. Each access control entry is composed of an ID for the user or group and the permission granted to that ID. The ID can be for a user in the local or trusted domain, or a host account. The NTFS permissions directly affect local file system access, and are compared with share permissions in the file sharing ACL for determining effective access to a share. "No Access" permission blocks all access to a file or directory, trumping any other permissions (local or shared). When granting users "Full Control" permissions, make sure they understand how to calculate effective share permissions:

1. Check for a "NO ACCESS" trump.
2. Check the most permissive SHARE ACL (they are a member of all groups).
3. Check the most permissive FILE ACL (they are a member of all groups)

Effective permission over a network is the most restrictive of the FILE and SHARE permissions. NTFS Version 1.1 (or NTFS Version 4.0) can be used by NT4, Windows 200x, and XP. NTFS 4.0 offers the following advantages over FAT:

- A more efficient allocation of space
- An ACL (multiple users and groups and multiple levels of access)
- Six permission groups – *no access*, *list*, *read*, *add*, *add and read*, *change*, and *full control*
- A static ACL inheritance model in which new files inherit from the folder they are in
- Support for very large files (a single file can use the entire partition)
- A journal of changes to the file system for auditing.
- Support for Redundant Array of Inexpensive Disks (RAID).
- Reduced fragmentation (compared to FAT), but it still occurs.
- Supports file names up to 255 characters and is case-sensitive.

NTFS 1.2 (or NTFS 5.0) became available with Windows 2000, and can be used with NT 4.0 SP4b, Windows 2000, and XP. The Security Configuration Manager (SCM) program must be installed in order for NT4 SP4b to use the new security features of NTFS 5.0. If a NTFS 4.0 partition is attached to a Windows 2000 system, it is converted to a NTFS 5.0 partition with the following enhancements:

- **Reparse Points** An object composed of metadata and an application filter (a program) stored with an associated file or directory on the file system. When the file is accessed, the metadata is passed through the filter, modifying how the file is accessed. A single file or directory can have multiple reparse points. There are some internal reparse points, and any NTFS 5.0-aware application can create them:
 - **Symbolic Links** Points to the real path of a file
 - **Junction Points** Symbolic links for the directory
 - **Volume Mount Points** Symbolic link for mount points
 - **Remote Storage Server** Rules to move files to offline or near-line storage; leaves a reparse point to get the file back.

- **New Security and Permissions Methodology (Dynamic Inheritance)** As the ACL of parent directories change, dynamic permission inheritance allows subdirectories to inherit the changes. Inherited permissions and manually set permissions are maintained separately. One drawback is that dynamic inheritance is more processor/memory intensive than static inheritance. “No Access” trump control has been removed since “allow” and “deny” controls were added. There are thirteen attributes to assigning permissions: *Traverse Folder/Execute File, List Folder/Read Data, Read Attributes, Read Extended Attributes, Create Files/Write Data, Create Folders/Append Data, Write Attributes, Write Extended Attributes, “Delete Subfolders and Files, Delete, Read Permissions, Change Permissions, and Take Ownership*. When determining effective permissions, integrate the following new rules:
 - “Manually set” wins over “inherited”
 - “Deny” wins over “allow”
 - “Parent inherited” wins over “grandparent inherited”
- **Change Journals** Logs all file system operations in a 64-bit update sequence number. The actual data changed is not stored.
- **Encryption** This option, called the Encrypting File System (EFS), stores data in an encrypted format. The OS creates 128-bit (or 40-bit) public/private keys; encryption is done with the public key and decryption is done with the private key. The encryption operations are part of the OS, not the file system. The EFS methods are part of the NTFS 5.0 hard and soft disk quotas based on users and groups and globally.
- **Sparse File Support** For data with long sequences of zeros, only non-zero data is written to disk, along with a record of where the zeros should be inserted. Once stored in this special way, the file is permanently changed and cannot be converted back to its original form.

Linux NTFS compatibility is limited to a *read-only* kernel driver or a *read-write* wine hack. The Linux NTFS kernel module has been reverse-engineered to allow a Linux system to mount and read an NTFS file system. At the time of this writing, the driver allows for writing the same length files as those already present on the disk, but a check disk (CHKDSK) is usually required by Windows NT, Windows 2000, and Windows XP upon reboot to clean up damage caused by the driver. Alternatively, the Captive project offers modules for *read* and *write*

access to NTFS partitions using wine's Microsoft Windows NT kernel emulation, by reusing one of the original *ntoskml.exe* ReactOS parts and the Microsoft Windows *ntfs.sys* driver. This is a reliable method for accessing an NTFS file system that requires NTFSPROGS v1.8.0 (a group of NTFS utilities based around a shared library), which requires GLIBC v2.3 (a GNU/Hurd and GNU/Linux library). The project home page can be found at: www.jankra-tochvil.net/project/captive/CVS.html.pl.

Since most network file services migrations occur by copying files across the network to the new Linux server, there is little need for an NTFS driver for Linux. However, in cases where there is a large amount (terabytes) of disk data, the network-based copying of files may take more than a day to complete, even over a fast network. In these cases it is often much faster to temporarily install the NTFS-formatted disks into the Linux machine and perform the file copy locally.

Understanding Linux File Systems

In addition to support for FAT and NTFS (mostly *read-only*), Linux also supports a multitude of other file systems. The 2.6 kernel natively supports second and third extended file system (ext2/3) and ReiserFS among others. Since ext2/3 and ReiserFS are the most commonly used file systems with Linux, they are the focus of this section.

Ext2/3 and ReiserFS have many common attributes. Every file and directory has an address on the disk that is referred to as an *inode*. They both support a concept of hard and soft links, where you can create a file name which points to another file name (similar in function to a Windows link) by using the inode value. A *soft link* is a file that contains the inode address of a file that stores the inode to the data. A *hard link* is a second file that contains the inode of the target data. Ext2/3 and ReiserFS are hierarchical, where every file or directory is stored in a directory with “/” or “root” at the top of the tree.

Linux file */etc/fstab* (file system table) lists file systems, mount points, and mount options, and is parsed when the system boots. When the file system is mounted, a line is added to file */etc/mstab* (mount table) and a “clean” bit in the partition header is read and then set to “0.” When unmounted (as it is during shutdown), the clean bit is set to “1.” During the boot process, if the “clean” bit is not set to “1,” the partition is referred to as “dirty” and special programs are run against the file system to check for errors. These programs (*fsck*, *fsck.ext2*, and *fsck.reiserfs*) perform “consistency” checks of data and metadata on the disk. The file system is “consistent” if each data block is either allocated to a single inode

or unallocated. Data blocks that belong to more than one inode are considered errors or an “inconsistency” on the disk, and represent lost or damaged data. If a “write” partially fails to complete, data in the file may be lost or information about block-file association may be damaged.

Understanding Ext2/3

Ext2 was the first file system used by Linux, and was based on the Minix file system. It offers ownership for user, group, and other, and permits *read*, *write*, and *execute* access control. File names are case sensitive and can be a maximum of 255 characters long. The maximum size of a partition or a file on a partition is 4 TB. One drawback to Ext2 is that files on the file system can be easily damaged by unclean dismounts (like a power failure) and required user intervention to reboot (including lengthy disk checks with file system check (FSCK), which can require user intervention to complete). When a file becomes damaged, it can disappear or become zero bites. If this happened to a critical file, the system could fail to boot.

Ext3 was originally developed by Dr. Stephen Tweedie at Red Hat. It was added to the release kernel beginning with 2.4.15, but is now used with 2.4.16 or newer. Ext3 can be thought of as an ext2 file system with journaling. To use ext3, you must have ext3 compiled into kernel and have the e2fsprogs suite installed. Ext3 offers the following three journaling modes that are set in */etc/fstab*:

- **Journal** Logs file data and metadata changes. This is the slowest mode.
- **Ordered** Logs file metadata changes and runs data updates to disk before updating metadata. This is the default mode.
- **Write Back** Logs file metadata but does not check data changes to disk. This is the fastest mode.

Converting a file system between ext2 and ext3 does not require that you back up and restore data like other file system conversions do. To convert an ext2 to ext3, add a journal such as:

```
/sbin/tune2fs -j <partition>
```

To convert an ext3 to ext2, clear the journal file system feature. For example:

```
tune2fs -O ^has_journal <partition>
```

Conversion back to ext2 is not necessary, because an ext3 file system can be mounted as either ext2 or ext3. Once an ext2 is converted to ext3, */etc/fstab*

should be edited to prevent FSCK from running the consistency check. Since there is a journal, the clean bit should not be needed to maintain file system consistency (in theory). In the */etc/fstab* entry, change the last column to a 0 to prevent the consistency check.

```
/dev/hda2  /data      ext2      defaults  1 2
/dev/hda2  /data      ext3      defaults  1 0
```

Resources

Kernel Parameters

- `CONFIG_EXPERIMENTAL=y` n # Needed in older kernels
- `CONFIG_EXT3_FS=y` n # Default in 2.6 kernel

Modules/Drivers

```
/lib/modules/2.6.5-7.104-default/kernel/fs/ext3/ext3.ko
```

Tools:

- **ext2/ext3** <http://e2fsprogs.sourceforge.net/ext2.html>
- **e2fsprogs** Contains utilities: `e2fsck`, `mke2fs`, `debugfs`, `dumpe2fs`, and `tune2fs`
- **e2fsck** Check file system for damaged or incorrect inodes, and allow for repair.
- **mke2fs or mkfs.ext3** Create an ext2/3 file system.
- **debugfs** Debug errors in damaged file systems.
- **dumpe2fs** Read the file system parameters.
- **tune2fs** Change the file system parameters.
- **ext2ed** The deprecated curses interface to edit ext2 file systems less than 2GB.
- **defrag** The defragmentation utility for the ext2 file system.
- **e2image** Create a regular or raw image file. `e2fsprogs` utilities work with raw files.
- **fdisk** The utility to view and edit partitions and file systems on a disk.

- **FSDEXT2** Win95 driver for *read-only* ext2 access
- **Ext2fsd** WinNT, Win2K, and XP drivers for ext2 file systems.
- **MountX** A Mac OSX driver for ext2 file system.

Understanding ReiserFS

Based on the work of Hans Reiser, ReiserFS was added to the 2.4 Linux kernel in 2001. At the time of this writing, ReiserFS v.3 ships with most Linux distributions, but v.4 was just released. The home page for the file system can be found at the Naming System Venture (www.namesys.com/). Versions are backward-compatible and if one version is converted forward (by mounting the “-o conv” option), utilities from the older version will not work. The “resize=<NUMBER>” remount option allows partitions to grow without having to back up or restore data; however, a partition cannot be shrunk. This is a true journaling file system, which keeps a ledger of transactions and transaction metadata with “roll forward” and “roll back” recovery. The process flow works like this:

1. Schedule transaction (if Step 1 fails, the write was lost)
2. Perform transaction (if Step 2 fails, the system can replay or remove the transaction)
3. Mark transaction complete (if Step 3 fails, the system will treat it as if Step 2 failed)

If the file system is left “dirty” (after a power failure), the journal is analyzed instead of running a time consuming FSCK, to check for inconsistency and replayed if needed. This can reduce downtime and the chance of a corruption.

ReiserFS organizes the file system into two areas: data and system. The data area consists of directories, files, and file metadata, organized as a single “balanced tree” data structure (in v.3) or a “dancing tree” data structure (in v.4). With a balanced tree structure for a given file, the data and metadata can be stored near each other on the disk, minimizing reader arm movement, which increases read speed. Locating a file on a partition is faster with a balanced tree structure than with the ext2/3 method, and is even faster when using a “dancing tree” structure. By using the exact block count storage method, less space is wasted than with the ext2/3 block-based allocations. The ReiserFS system’ area consists of super block, journal, and bitmap. ReiserFS journal can repair bad blocks in the

data area, but not in the system area. The system area does not normally have corruption, but can be repaired by rebuilding the super block or tree. For example:

- `/sbin/reiserfsck -fix-fixable --rebuild-sb /dev/hda2)`
- `/sbin/reiserfsck -fix-fixable --rebuild-tree /dev/hda2)`

Kernels prior to 2.4.7 had a problem with NFS shared ReiserFS partitions, including file system corruption and failed writes, but this seems to have been fixed. In the beginning, there were reported problems with software RAID, but hardware RAID has always worked. With a combination of JFS and ReiserFS, software RAID can be achieved.

ReiserFS v.4 boasts the fastest file system, is atomic (transaction corruption does not occur), offers more efficient file storage (replacing balanced tree algorithm with dancing trees), is extensible with plug-ins, and has military grade code (with assertions entering each function).

Unlike ext2 and ext3, many kernel parameters are available for ReiserFS, for compiling a new kernel (not covered here) to enable special features of the file system. To build ReiserFS support into the kernel, set `CONFIG_REISERFS_FS` to “y;” to build the module, set it to “m.” If `CONFIG_REISERFS_CHECK` is set to “y,” the file system will run in debug mode, will run every check possible, and will perform slower; therefore this is normally left at “n.” Setting `CONFIG_REISERFS_PROC_INFO` to “y” will create a larger kernel/module, will require more kernel memory, and file system statistics will be stored in `/proc/fs/reiserfs`. A kernel patch is available that will add an option called `CONFIG_REISERFS_RAW`. If it is set to “y,” it will allow raw access to the ReiserFS internal tree, bypassing the file system. See Kernel source *Documentation/filesystems/reiserfs_raw.txt* for further information. Kernel option `REISERFS_HANDLE_BADBLOCKS` becomes an option if you edit the kernel source and include `/linux/reiserfs_fs.h`, which is used to locate and mark bad blocks on a mounted ReiserFS partition.

Resources

Modules/Drivers

`/lib/modules/2.6.5-7.104-default/kernel/fs/reiserfs/reiserfs.ko`

Tools/Utilities

- Get information about a ReiserFS partition (version, block size, and so on)

```
debugreiserfs <partition>
```

- Generate a list of bad blocks on a ReiserFS partition:

```
/sbin/badblocks [-b <reiserfs-block-size>] <partition>
```

- Run consistency checks and fix file system errors:

```
reiserfsck --check <partition>
```

- Fix bad blocks on a ReiserFS partition:

```
reiserfsck --badblocks <file with list of bad blocks> <partition>
```

NOTE

fsck.reiserfs and *reiserfsck* are the same program.

- Create a ReiserFS file system:

```
mkreiserfs <partition>
```

NOTE

mkfs.reiserfs and *mkreiserfs* are the same program.

- Modify the journal size or the maximum transaction size:

```
reiserfstune <partition>
```

- Mark blocks as bad in the file system journal:

```
reiserfstune --badblocks bad_blocks.txt /dev/hda2
```

Understanding Permissions Management (Access Control)

File systems NTFS, ext2, ext3, and ReiserFS all support access control in different ways. Windows shares and Samba shares also address access control in different ways. The goal in migrating from Windows to Linux is to maintain as much of the ACL as possible when the shares are moved to a Samba server, and over time try to simplify the ACLs back to the owner, group, and other UNIX style. This section is concerned with the end user access to network shares. In order to cover this, we must first cover basic UNIX permissions and Extended Attributes (EAs)/ACL features in Linux.

In the simple form (and in historical UNIX) file-based access control is called the “workgroups system” and consists of only one level and three types of ownership per file or directory (one user, one group, and world) and three types of access control (*read*, *write*, and/or *execute*). User and group identity is traditionally set in */etc/passwd* and */etc/group*, although user ID (UID) and group ID (GID) can be attained from other areas such as NIS or OpenLDAP. Ownership changes are performed with the commands *chown* and *chgrp* and are controlled with permission bits: nine file mode bits (*read*, *write*, *execute*) and three special bits (*setuid*, *setgid*, and *sticky*). The command *chmod* can be used to set the permission and special bits, but some commands specifically set the three special bits (*setuid*, *getuid*, *setreuid*, *seteuid*, and *setfsuid*). On a directory, *read* allows you to list the contents, *write* allows you can create files within, and *execute* allows you to search a directory.

Linux also has an answer for cases in which more complex file-level permissions are required. Through patches to the file system code in the kernel (now incorporated in current kernel releases), the full-featured POSIX ACLs and EAs become available. POSIX ACLs can be used when standard UNIX permissions are not granular enough. Permissions can be assigned to users (or groups) on an individual basis (i.e., Joe can have *rx* to *foo.txt*, while Sally can have *rw* to *foo.txt*). EAs allow you to control how a file or directory will interact with the file system, causing a file to become immutable, a directory to compress its contents, or a journal any writes in a particular way.

Different permissions can be specified for many users and groups. ACL's have been a part of Windows NT, AIX (IBM Unix OS), and HP-UX and has been available as a patch to the 2.4 Linux kernel, but EA/ACL is included in the 2.6

kernel for ext2, ext3, and XFS. A patch to the 2.6 kernel is still needed to add EA/ACL support for ReiserFS, but the SuSE 2.6 kernel includes this patch.

Linux POSIX ACLs add NT-style permissions to Linux. Commands *getfacl* and *setfacl*, which use the *libacl.so.0* shared library and are part of the *e2fsprogs* package, allow greater permission granularity. Different permissions can be assigned based on several different UIDs or GIDs instead of just one UID, one GID, and an Other default. EAs can be accessed with the commands *lsattr* and *chattr* (also part of the *e2fsprogs* package), allowing a user to control how the file system will allow access to a file.

ACL attributes are inherited from the parent directory at the time of creation. An ACL entry is composed of: 1) a “type” (user, group, mask, or other); 2) a user/group or blank; and 3) permissions. When no user is specified for a given type “user” or “group,” permissions apply to the user or group that owns the file or directory. Permissions resolution is determined in the following non-cumulative order: owner, named user, owning group, named group, other. Table 5.1 shows the terms given to the following types of ACL entries.

Table 5.1 ACL Entry Terms

Term	Text Form	Note
owner	user::rwx	traditional UNIX owner
named user	user:root:rwx	
owning group	group::rwx	traditional UNIX group
named group	group:admin:rwx	
mask	mask::rwx	
traditional UNIX other	other::rwx	traditional UNIX other

Effective permissions are calculated by performing a logical AND between the user and mask entry. See Table 5.2 for an example.

Table 5.2 ACL Types, Entries, and Permissions

Type	Entry	Permission
named user	user::rw-	read and write
mask	mask::r-x	read and execute
Effective permission	r—	read

After the “access ACL” is set on a file, a long listing will show a plus (+) sign after the permission bits.

```
i.e. ls -ld foo
drwxrwx--- 2 jstile users 48 2004-08-24 06:18 foo
setfacl -m user:root:rwx,group:root:rwx foo
ls -ld foo
drwxrwx---+ 2 jstile users 48 2004-08-24 06:18 foo
getfacl foo
# file: foo
# owner: jstile
# group: users
user::rwx
user:root:rwx
group::r-x
group:root:rwx
mask::rwx
other::---
```

When ACL feature “default ACL” (using the *-d* option) is set on a directory, the directory retains its ACLs, but new files and directories created under that directory will inherit “default ACL”.

```
Set the default ACL on the directory 'foo':setfacl -d -m group:ntadmin:r-x
foo
getfacl foo/
# file: foo
# owner: jstile
# group: users
user::rwx
user:root:rwx
group::r-x
group:root:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:ntadmin:r-x
```

```
default:mask::r-x
default:other:----
```

To remove an ACL, use the `-x` option:

```
setfacl -d -x group:ntadmin foo
getfacl foo/
# file: foo
# owner: jstyle
# group: users
user::rwx
user:root:rwx
group::r-x
group:root:rwx
mask::rwx
other:----
default:user::rwx
default:group::r-x
default:mask::r-x
default:other:----
```

NOTE

Only the UNIX owner/group or root can delete or move the file, even if the ACLs grant `rwx`.

Only the UNIX owner/group or root can modify the ACL, even if the ACLs grant `rwx`.

`setfacl` and `getfacl` do not list or change `setuid`, `setgid`, or sticky bits.

`/etc/fstab` must mount the file system with the ACL option, or POSIX ACL commands will fail (i.e., `/dev/hda2/ext2 acl,defaults 1 1`).

`chmod` will change the ACL mask, and the ACL grant must be reapplied or the mask must be reset to `rwx`.

If a program's permissions are `4700` and a user is granted ACL `rwx`, the program will not run as `setuid`.

You can create a backup of EA/ACL file system metadata with the following command:


```
getfacl -R --skip-base / > /root_AE_ACL_BACKUP_`date`.acl
```

You can restore ACL's from a backup file with the following command:

```
setfacl -restore=<your backup file>
```

When Samba is configured to use PAM (Pluggable Authentication Modules), PAM is configured to use OpenLDAP, and the Name Service Switcher (NSS) is configured to use OpenLDAP and wins, Lightweight Directory Access Protocol (LDAP) accounts can be used to assign ownership to files and to authenticate domain users as a Samba primary domain controller (PDC). When patched ext2, ext3, or ReiserFS file systems have been mounted with the “acl” option, and the Samba-3 *smf.conf* file contains “map acl inherit = Yes,” shares and files on the file system can be configured with POSIX ACLs. If you configure the Samba-3 server as a domain controller (DC) with the OpenLDAP backend and winbind, domain users and groups can be given permission to share the way it is done with Windows file servers. This might make Samba look very much like a Windows server to a Windows administrator, which is good and bad. On one hand, it allows the administrator to migrate ACLs to shares as they are set on a Windows server, easing the migration path. On the other hand, two big problems arise by using POSIX ACLs: 1) they are hard to move between machines; and 2) they are difficult to back up or restore. Drag-n-drop of files from one server to another with the Windows Explorer does not preserve POSIX ACLs, and neither do Linux tools such as SCP (Secure Copy command), RSYNC (a command that allows fast incremental file transfer), and TAR (derived from Tape Archiver). The recent version of the *cp* command preserves ACLs with the “-p” option, if copied to a locally mounted file system that is mounted with the “acl” option. Alternatively, you can also use “star,” which is a POSIX-aware version of tar (<ftp://ftp.berlios.de/pub/star/>). To create a tar of a directory including the ACLs:

```
star H=exustar -acl -c <path> > archive.tar
```

To restore the archive, including ACLs:

```
Star -acl -x < archive.tar
```

The *H=exustar* has star create an extended pax archive. The POSIX 1003.1-2001 standard defines pax as an archive format compatible with the tar format with the addition of ACLs. From Windows, you can use robotcopy (from the Windows XP Resource Kit), xcopy with xacfs (from the Win2k Resource Kit), or scopy (from the NT Resource Kit) to transfer ACLs, but Microsoft resource kits are not free. The long-term solution is to simplify our ACL's as you transfer

shares to Samba, by trying to change your security model to use standard UNIX file permissions (one owner, one group, and other) with a combination of file system and share controls. With file system permissions, you can set the sticky bit on a directory to ensure that only the owner of a file in that directory can delete or rename the file. Setting the UID or GID bit on a directory ensures that all files created in that directory will be owned by a set user or group ID. In concert with file-based permissions, the share configuration in *smb.conf* allows the administrator to grant various accesses based on user and group. Table 5.3 is part of Samba's HOWTO Collection, and identifies user- and group-based controls.

Table 5.3 User and Group-based Controls

Control Parameter	Description
Admin users	List of users with full control of anything in this share.
Force group	Treat anyone using this share as if a member of this group.
Force user	Treat anyone using this share as if this user.
Guest ok	No authentication required to access this share.
Invalid users	List of users to be blocked from accessing this share.
Only user	???not sure about this one???Do not allow users to access share if they are not in the admin list.
Read list	List of users allowed to only read items in this share.
Username	???
Valid users	List of users allowed to access this share.
Write list	List of users allowed to read and write to this share.

Understanding File Backup, Restore, and Replication Options

File services and storage cannot be discussed without talking about backup, restore, and replication. Since the files of a company typically contain valuable intellectual property, these assets must be appropriately protected, and contingency plans must be planned for problems ranging from the accidental deletion of a file or directory to the complete destruction of a server and disks.

To properly protect against (or, more likely, prepare for) these types of events, it is necessary to back up the data to removable media for safekeeping.

Replication may be one-time, scheduled, or continuous. Backup of the OS files will allow for more rapid disaster recovery. Best practices dictate that at least some of the backup media is stored offsite. If feasible, backup media stored onsite should be stored in fireproof safes.

While rewriteable optical media is useful for one-time and small job backups, the de facto choice for corporate backup needs in the range of tens, hundreds, and thousands of gigabytes of data is tape media.

For most system administrators, it's not a case of *if* you have to perform a restore, but *when* you'll have to perform a restore

Having backups of data on tape is useless if you can't find the data you need. Just as planning for a backup is important, planning for file (and possibly server) restore services is equally important. It is a best practice to perform test restores from tape at least monthly. The time to find out that the data being backed up to tape isn't restorable because of a bad tape, incorrect formatting, or other errors is before a disaster strikes, not after. A backup strategy attempts to answer five questions:

1. What files should be backed up?
2. How should they be backed up?
3. What type of media is appropriate?
4. When should backups run and which types?
5. When should a backup be discarded?

Identify Critical Files to be Backed Up

Some files are hard to replace, such as company intellectual property, configuration files, log files, home directories, and custom compiled programs. Typically, these types of files are located in `/etc` `/var` `/home` `/root` and `/opt`. Other files are easy to replace, such as standard distribution packages, temporary kernel files, or device files, and should not be backed up. These are typically part of `/dev`, `/proc`, and files supplied by a standard Linux install. Choosing to back up only necessary items will minimize the size of each backup and reduce the resource cost in designing a backup solution. Grouping critical files in common locations will also simplify the process of deciding what to back up.

It is also good to have a clone of each type of computer on hand, to quickly generate new machines. Clone images should be kept in near-line storage (like a hard drive), as tape media can be very slow. If a drive fails, the clone can be placed on a new disk in a short period of time, and critical files can then be restored from tape in less time than it would take for a full backup restore from tape. System Imager (part of the System Installer suite) is a great distribution-independent cloning solution.

Decide on a Backup Method

There are many ways to design a solid back method. Some suggestions include:

- Run a backup on each host to some type of backup media on each host. This may work for a small number of machines, but it is not a scalable solution.
- Copy all of the important files to a backup server (with *rsync*), where it will be written to backup media with a command line utility.
- Backing up many machines can be accomplished without any special software.
- Set up a backup client/server suite. In this solution, each client to be backed up runs a daemon and the server with backup media runs a server daemon.
- The server daemon will communicate with each client daemon, transferring data to the backup media (i.e., for Advanced Maryland Automatic Network Disk Archiver (AMANDA), HP Omni Back, ARKEIA, Veritas NetBackup).

Determine What Type of Backup Media to Use

As media types increase in capacity and transfer speed, they generally increase in cost. The media and drive price is the largest upfront cost in most backup solutions, and will also have additional annual costs. Always buy media in bulk (at a reduced price), as it will decrease the long-term costs for media. If possible, obtaining a library (a multi-cassette drive) will help reduce the overhead of swapping tapes. Another consideration is that some backup media, while durable, will become obsolete because people will not have a device to read the media. Examples include reel-to-reel tape, exabyte, jaz, zip, and Magneto Optical. When

choosing an archive media, consider archiving a drive that can read the media, and a driver for the drive to avoid inevitable obsolescence. Alternatively, include a transfer of old archives to the new media type in your backup solution upgrade strategy. This will add realistic additional costs to any upgrade, and will save the administrator from having inaccessible archives.

Media types commonly used include compact disk (CD)/Digital Versatile Disk (DVD), disk drives, Digital Audio Tapes (DAT), Digital Linear Tapes (DLT), and Linear Tape-open (LTO). CD/DVD media (about 4GB capacity) is cheap and can be read by a machine with a CD/DVD reader, which are common in modern PCs. There are several drawbacks to CD/DVD media. Organizing the CD's becomes bulky and requires excellent organization. CD or DVD writers need a new blank disk each time a backup is performed. The cost of media depends on how many backups are performed and the number of disks per backup. It has been discovered that media deteriorates over time, leading to unusable archives. In addition to scratches, the glue can separate, the plastic can become opaque, and the silver backing can flake off. Large capacity disk drives can transfer data quickly, and offer a cheap media type (in terms of data per dollar).

Firewire drives have become very popular; some feature films are archived in this format instead of on film. The drawback is that there are mechanical parts that can fail, and drives are sensitive to impact. DAT tapes (up to 20 to 40GB) were the media of choice years ago.

Tape media is durable, small, and rewriteable, and a multi-cassette changer is relatively cheap. However, they are also very slow and relatively low in capacity.

DLT tapes (up to 40 to 80GB) are popular media in current backup solutions. They have a faster transfer rate (between 1.2 and 16 MBps), hold more data, are durable, and are rewriteable. However, the media and drives are more expensive.

LTO Ultrim's (up to 100 to 200GB) are popular for medium to large networks. They transfer at a very fast rate (20 to 40 MBps), are about the same size as a DLT, and are rewriteable and durable. The problem is that they are very expensive.

Backup Schedules

Backup schedules tend to include full and incremental backups. A full backup ensures that you can restore a file using only the latest backup media. Full backups take more time to perform and use more media than an incremental backup, but restoring from a full backup takes less time. Incremental backups

ensure that a file can be restored using media from the last full backup and several increments. Consider how long you are willing to wait for a restore request and how long a full backup takes when planning your schedule. The more frequent full backups are scheduled, the greater the annual cost for media. The backup schedule should be agreed upon by the managers and administrators in order to find a happy medium between cost and expectations. It will require more full backups (more media and drives) if management wants a service level agreement of one-hour restores.

Keeping a *calendar* allows you to easily see when backups have been performed and when they are next scheduled to be performed. It also helps to keep a journal of details about each backup. This data can be kept in a database or flat file. Some commercial solutions offer their own method. It allows administrators and users to quickly determine which tape contains the desired material, and also shows failed backups.

Plan to Rotate and Archive Media

A best practice is to archive some, but not all, full backups for long-term storage. As a backup gets older, the cost of media storage increases, data on the media becomes obsolete, and its usefulness diminishes. A trade off with removing backups is a matter of granularity: only one backup will represent a larger window in time. Many forms of media can be erased and reused. To prevent the steady rise in new media for each backup, media from older backups can be erased and reused. When reusing media, take care to note how many times the media has been used (via a media serial number or physically marking the media). Everything breaks down over time; you don't want to trust a backup (your safety net) to damaged or faulty media.

Backup all extended attribute ACLs to a file:

```
getfacl -R --skip-base / > /backup.acl
getfattr -dhR -m- -e hex / > /backup.ea
Restore the ea/ACL's:
cd /
setfacl --restore=backup.acl setfacl --restore=backup.acl
```

Backup the *mysql* database:

```
mysqldump --tab=/path/to/some/dir --opt db_name
```

For Linux device files, the Small Computer System Interface (SCSI) tape drives, the “rewind” device is typically */dev/st0*, “no rewind” is */dev/nst0*. With a

SCSI library changer drive is `/dev/sg0`, and library is `/dev/sg1`. For Integrated Drive Electronics (IDE) tape drives the “reqind” device is `/dev/ht0` and “no rewind” is `/dev/nht0`.

Many command-line programs exist for compressing file(s) and flexsystems, and for writing to backup media. These include Copy In and Out (*cpio*), *tar* (or GNU *tar*), *dump/restore*, *mt*, *dd*, and *cdrecord*. On their own, these programs can be used to create a low-tech backup/restore solution. The *mt* command is used to *operate*, *read*, *seek*, and *write* to a magnetic tape, while the *xmt* and *loaderinfo* commands are used to operate a library changer (holds multiple tapes).

The following examples show how to use *cpio*, *mt*, and *xmt*:

- Rewind and eject a tape:

```
mt -f /dev/st0 rewind
mt -f /dev/st0 eject
```

- List tape contents:

```
cpio -itv -I /dev/st0
```

- Backup the `/data` directory to tape with *cpio*:

```
cd /data
find . -print | cpio -ovH crc -O /dev/st0
```

- Backup the `/data` directory to tape with *tar*:

```
cd /data
tar -clpMvf /dev/st0 *           # backup directly to tape.
tar -dMf /dev/st0               # verify content
```

- Backup everything with *tar*, except `/proc` and `/dev`:

```
tar -cpfM /dev/st0 / --exclude=/proc,/dev
```

- Backup with *dump*:

```
dump 0f /dev/nst0 /
```

- Restore data from tape with *cpio*:

```
cpio -icvmuld -I /dev/st0
```

- Restore everything from tape with tar:

```
tar -xpf /dev/st0 -C /
```

- List tapes in a library:

```
mtx -f /dev/sg1 inventory
```

```
mtx -f /dev/sg1 status
```

- Load tape 1 in a library to the tape drive:

```
mtx -f /dev/sg1 load 1 0
```

A better solution is to use an open-source backup suite such as Mondo, StoreBackup, or AMANDA. These programs are wrappers around the previously mentioned shell commands, but they also offer extra features such as logging, scheduling, and so on. Mondo is a CD-based backup solution for a single host and StoreBackup is a disk-to-disk backup solution for a single host. Mondo uses *cdrecord* to create a bootable CD and a number of data CDs. StoreBackup makes a clone on another disk.

AMANDA

AMANDA is a client/server suite with a shell interface for automating the backup of multiple networked hosts to a single host with tape drive(s). It can be found at <http://AMANDA.org>, with a current release of 2.4.4p3. Originally written by John R. Jackson and Alexandre Oliva, AMANDA offers backup schedules, logging, multi-drive/tape backups, and concurrent remote host backups. It will attempt to keep a backup running even if there are tape problems. AMANDA can use tape changer libraries, and can run full or incremental backups in the same way the dump command works.

The dump methods are:

- Level 0 is full
- Level 1 is changed since last level 0
- Level 2 is changed SINCE last level 1

AMANDA keeps backup logs, tracks tape usage, and print tape labels, With the shell script, *amplot*, AMANDA can generate a graphical display of your backup. The *dumpcycle* is how often AMANDA performs a complete -f0-

type dump. With multiple network cards in an AMANDA server, network traffic can be directed to use a specific interface.

AMANDA Backup Process

AMDUMP runs from cron (the daemon used to execute scheduled commands) on the AMANDA tape server as user “AMANDA-user” and group “backup,” pausing if a file named “hold” exists in the domain *config* directory (*/opt/AMANDA/etc/MyCompany.com/hold*). If no file named “hold” is found, *amcleanup* is run if the last backup was aborted or did not finish clean, which writes any holdingdisk data to tape and then runs *amflush*. To erase all data from a previous failed backup, run *amflush* by itself and then *planner* querying clients and schedules on the appropriate level of backup. The driver program is started with a taper process (one *reader* thread and one *write* thread) and one dump process for each line in the schedule. Each process dumps a backup from a client to the holding area or directly to tape.

Finally, the *amreport* script renames the log report and e-mails a report, and *amtrmidx* updates the indexing.

AMANDA runs on UNIX hosts and can back up Microsoft Windows hosts using Samba and kerbrose.

When backing up an MS Windows host, you must remember to dump the registry to a file that can be backed up with the Windows NT Resource Kit utility *regback.exe* and *regrest.exe* to restore, which is not an open source.

The following batch file will back up a registry:

```
regback.bat
del C:\regbkp\my_host_name\*. * /q
regback C:\regbkp\my_host_name
```

It should be noted that a database change in v.2.4.0, leaves pre-v.2.4.0 backups unreadable in the current version.

To resolve this problem:

1. Upgrade the pre-v.2.4.0 databases.
2. Export with a pre-v.2.4.0 client.
3. Import the database with a <v. 2.4.0 client.

AMANDA uses *dump* and *tar* to generate backups, and the “no rewind” */dev/nst0* device when writing to tape.

The most effective way for AMANDA to operate is by dumping a backup to a “holdingdisk” (a scratch directory) before streaming data to tape. This has three advantages:

1. Since a *dump* to disk is typically faster than a *write* to tape, the *write* to tape will be a constant stream, reducing wear on the drive and the tape.
2. Multiple dumps can occur concurrently, shortening the backup window involving remote hosts.
3. If the system runs out of tape, backups can continue writing to the holdingdisk, and eventually be flushed to tape at a later time when more tapes are loaded.

If the system reboots or the *write* to tape fails, you must run `amflush` to write the holdingdisk to tape.

AMANDA has a configuration tool for determining the optimum settings for a tape drive:

```
amtapetype -t "BNCHMARK DLT1" -f /dev/nst0
```

The tool takes a very long time, but determines the proper settings for your drive, media, and SCSI configuration. The tool is only necessary if you have a drive that is not listed in the default *AMANDA.conf*.

The output with an Adaptec 2940UW card (transferring at 20.000MB/s), a BNCHMARK DLT1 drive using a DLT IV (40 to 80GB) cartridge, and an `amtapetype` (a generated `tapetype` definition) that is run for 13 hours is:

```
amtape -t "BNCHMARK_DLT1" -f /dev/nst0
Writing 256 Mbyte   compresseable data:   35 sec
Writing 256 Mbyte uncompreseable data:  118 sec
WARNING: Tape drive has hardware compression enabled
Estimated time to write 2 * 1024 Mbyte: 944 sec = 0 h 15 min      #<--
took 15 minutes
wrote 980019 32Kb blocks in 2997 files in 22248 seconds (short write)
#<-- took   6 hours
wrote 950290 32Kb blocks in 5830 files in 28447 seconds (short write)
#<-- took   7 hours

define tapetype BNCHMARK_DLT1 {
comment "just produced by tapetype prog (hardware compression on)"
length 31604 mbytes
```

```
filemark 335 kbytes
speed 1239 kps
}
```

The output of `amtapetype` is a tape-type descriptor, which must be added to *AMANDA.conf*.

Installing AMANDA

You probably have some choices for installing AMANDA. Debian and Gentu use *apt-get*:

```
Find it:
    apt-cache search AMANDA
Install:
    apt-get AMANDA
SuSE, Mandrake, and Red Hat use rpm packages.
    rpm -i AMANDA-1.4.4p3.rpm
```

If you need a special patch, a heterogeneous UNIX environment, or require special compile options, you must build AMANDA from scratch:

1. Download the latest release from
<http://www.AMANDA.org/download.php>:

```
wget http://easynews.dl.sourceforge.net/sourceforge/AMANDA/AMANDA-2.4.4p3.tar.gz
```

2. Move the tarball (an archive of files created with the Unix tar utility) to a build directory:

```
mv amanda.1.4.4p3.tar.gz /usr/src/
pushd /usr/src/
    Unpack
tar -zxvpf amanda.1.4.4p3.tar.gz
cd amanda.1.4.4p3
```

3. Make directories for AMANDA:

```
mkdir -p /opt/amanda/{doc,etc,lib,libexec,man,share}
chown -R amanda-user.backup /opt/amanda
Build AMANDA:
```

```
./configure --prefix=/opt/amanda \
--with-config=MyCompany.con \
--with-user=amanda-user \
--with-group=backup \
--with-owner=amanda \
--with-smbclient=/usr/bin/smbclient \
--with-fqdn
```

The `—prefix=<directory>` command installs all of the AMANDA files into one place. The `—with-config=<name>` command creates a sample *AMANDA.config* with the `<name>` domain.

The `—with-user`, `—with-group`, and `—with-owner` commands specify the user and group AMANDA will run as.

The `—with-smbclient=` command allows AMANDA to back up MS Windows clients. The `—with-fqdn` command allows AMANDA to use fully qualified host names.

NOTE

If compiling for a client that will never run as a server, add the `—without-server` option.

To compile AMANDA:

```
make
```

Switch user to root and add the `amanda` group and user.

```
su root
```

```
groupadd -g 37 backup
```

```
useradd -u 37 -g backup -d /opt/amanda -c 'AMANDA admin' -s /bin/false -k
/dev/null -m amanda-user
```

```
Install
```

```
make install
```

The following directories will be created in `/opt/amanda`:

```
sbinAMANDA server side programs
```

```
libexecAMANDA backup client programs
```

```
lib AMANDA dynamic libraries
```

```
man      Directory for manual pages
etc/example  Sample configuration files, including AMANDA.conf and disklist
doc       Documentation
contrib    Extra scripts
```

NOTE

If you need to uninstall or remove all of the created files, do so from the build directory run.

```
make uninstall
make clean
rm -rf /opt/amanda
```

On the Server

After installing AMANDA:

1. Copy extra data from the source tree to the AMANDA install directories:

```
mv amplot contrib docs /opt/amanda/
mv example /opt/amanda/etc/
chmod +x /opt/amanda/amplot/amplot.sh
```

2. Create a directory for the holdingdisk:

```
mkdir -p /dumps1/AMANDA
```

3. Fix ownership issues that may have changed:

```
chown -R amanda-user.backup /opt/AMANDA
chown -R amanda-user.backup /dumps1
```

4. Add AMANDA to */etc/services* on the tape server:

```
amanda          10080/udp
amandaidx       10082/tcp
amidxtape       10083/tcp
```

5. Edit the *xinetd* file for AMANDA:

```
vi /etc/xinetd.d/amandaidx
# default: off
# description: AMANDA backup server with indexing capabilities
#
service amandaidx
{
    socket_type      = stream
    protocol         = tcp
    wait             = no
    user             = amanda-user
    group            = backup
    server           = /opt/amanda/libexec/amandad
    disable          = yes
}
```

6. Make a directory for the backup domain:

```
mkdir /opt/amanda/etc/MyCompany.con
```

7. Create an *amanda.conf* file by copying the example *amanda.conf* file into the backup domain directory.

```
cp /opt/amanda/etc/example/amanda.conf
/opt/amanda/etc/MyCompany.con/
```

Read and edit *amanda.conf* as needed. The *amanda.conf* file holds most of the configuration information for AMANDA. These settings include the user/group it runs as, the kerberose settings, the tape drive settings (tapetype), the changer library settings (tpchanger), temporary disk storage (holdingdisk), a catch all for most backup options (dumptypes), and a network interface card (NIC) for backup (interface). There are two main areas to the *AMANDA.conf* file: holdingdisk and dumptype. Holdingdisk definitions define a location for backup data to be stored before writing to disk. It is recommended to configure multiple holding disks for simultaneous backups, with enough space to accommodate one backup per holding disk, since holding disks are not earmarked for specific backups. If a holdingdisk with enough area to hold the backup is not available,

the backup will be written directly to tape and additional backups will wait until the drive becomes available. The following is an example of a holding disk entry:

```
holdingdisk hd2 {
directory "/dumps1/AMANDA"
use 30000 Mb
}
```

The next big area of *amanda.conf* is the dumptype definitions. There can be many dumptypes definitions in a single *amanda.conf*, and a dumptype definition can be used in other dumptype definitions as a method of consolidating common settings. The following is an example dumptype definition with some description:

```
define dumptype mother {
auth    bsd      # bsd/krb4
comment "my  backups" # description
#comprate      0.50, 0.50    # compression rate
compress       none    # none/client best/client fast/server best/server fast
dumpcycle      30      # days between full dumps
exclude "./dev/"      # file or pattern to exclude
exclude "./proc"      # file or pattern to exclude
holdingdisk    yes     # use holding disk (yes/no)
#ignore        # don't use this backup.
index no       # keep index (no/yes)
kencrypt       no      # encrypt transfer(no/yes)
maxdumps       1       # concurrent dumps on client. Default: 1
maxpromoteday  0       # Default: 10000
priority       high
program "GNUTAR"      # DUMP/GNUTAR
record yes      # record dump to /etc/dumpdates.
skip-incr      0       # Skip the disk when dump 0 is not due.
}
```

Configure a backup by creating a “disklist” configuration file in the backup domain directory (*/opt/amanda/etc/MyCompany.con/*). Each line in the disklist file represents one device or share to back up. Each line should contain a fully qualified domain name (FQDN) of the client to be backed up, a device/partition to backup, and a dumptype. The format of a line in the disklist is:

```
hostname diskdev dumptype [spindle [interface]]
```

Here are a few examples of disklist syntax:

- To back up */home* on the AMANDA server:

```
localhost          sda1          nocomp-root
```

- To back up a Windows share named `\\Host\C`, specify a Samba server as the host name:

```
samba_host.domain.com  "//ms_window_pc/C$."  nocomp-root
```

- To back up */home* on another Linux host:

```
linux_host.domain.com  /home          no comp-rootmkdir -p
/opt/sbin/amanda/MyCompany.con/amdump
chown -R amanda-user.backup /opt/amanda/MyCompany.con
```

On Linux Clients

Linux clients should consider the following steps:

1. Let user *amanda-user* connect from the AMANDA server:

```
echo '' <<EOF > /opt/amanda/.amandahosts
192.168.0.221  amanda
```

EOF

2. Set ownership and permissions:

```
chown amanda-user.backup /opt/AMANDA/.amandahosts
chmod 0400 /opt/amanda/.amandahosts
```

3. If you have *xinetd*, configure it and start:

```
vi /etc/xinetd.d/amanda
# default: off
# description: amanda backup client
#
service amanda
{
    socket_type      = dgram
```



```

        protocol      = udp
        wait           = yes
        user           = amanda-user
        group          = backup
        server         = /opt/amanda/sbin/amandad
        disable        = yes
    }
    /etc/init.d/xinetd restart
    If you have inetd, configure and start it.
    vi /etc/inetd.conf
        amanda          dgram    udp      wait    amanda
/usr/lib/amanda/amandad  amanda
    /etc/init.d/inetd restarted

```

For Windows Clients

Add the *share* password to the */etc/amanda* password on the Samba server that will be used to mount the Windows client.

AMANDA will contact the Samba server who in turn will contact the Windows client.

Set Up a Backup on the Server

AMANDA users must load a blank tape into the drive and write a label to it. AMANDA backups will not use a tape that does not have a label.

```
amlabel MyCompany.con Label1
```

1. Check the *config* and fix any problems that appear.

```
amcheck MyCompany.con
```

The output will look like this:

```

AMANDA Tape Server Host Check
-----
Holding disk /dumps1/AMANDA: 51309004 KB disk space available, that's
plenty
ERROR: cannot overwrite active tape Label1
      (expecting a new tape)
NOTE: skipping tape-writable test

```

```
NOTE: info dir
/var/lib/amanda/MyCompany.con/curinfo/motherin.MyCompany.con/_: does not
exist
NOTE: index dir / /amanda/MyCompany.con/index: does not exist
Server check took 6.408 seconds
AMANDA Backup Client Hosts Check
-----
Client check: 1 host checked in 0.312 seconds, 0 problems found
```

2. Run the backup manually:

```
amdump MyCompany.con
```

3. Cron AMANDA for automated backups:

```
# checks the tapes
0 16 * * 1-5 /usr/sbin/amcheck -m /etc/AMANDA/MyCompany.con/amanda.conf
# runs the backup
45 0 * * 2-6 /usr/sbin/amdump /etc/AMANDA/MyCompany.con/amanda.conf
```

Designing Linux-based File Services

Designing Linux-based file services is one of the most visible areas of a network design because it directly impacts end-user experience. Users will complain if their home directory is located across a wide area network (WAN) link, on a slow (or oversubscribed) server, or if a file suddenly becomes unavailable or they can't save their documents. Because an optimum design is specific to each scenario, it is best to follow some general rules. Optimizations take into account network topology (slow WAN, subnets, geography), hardware (disk, network, memory, CPU), redundancy (replication of the data), and user profile configuration (roaming profiles, non-synchronized files with folder redirection).

Hardware Resources

Traffic flows from the NIC to the memory to the disk. The slowest point along this path will be the bottleneck. Plan on requiring 100 Kbps per user and a profile directory of about 200MB per user (depending on how well profiles are locked down). The success of a new Samba server is directly related to how well you estimate required capacity and throughput.

Network speeds keep improving and users files keep getting bigger each year. When setting up your DC, estimate 0.1 Mbps per user to the DC where the user's home directory is located, and double that number for estimated peak usage. If the DC is also serving other files, allocate even more bandwidth. Assuming 100 Kbps, 30 users will pull 3 Mbps from the server. Plan on installing at least 1GB Ethernet cards on the DC for efficient performance. If more than one card is used, consider "channel bonding" (or as Intel calls it, "teaming") to bind the cards as a single interface, and configure the switch to handle this as well. Check hardware auto-negotiation between NICs and switches or routers, which often fail to choose the optimum settings, resulting in bad performance. Compare your network performance with hard-coded vs. auto-negotiated network settings for duplex, speed, and flow control.

Disk access must be at or near the network bandwidth to handle the input/output (I/O) requests, or your load will be very high (due to waiting processes). With a standard Peripheral Component Interconnect (PCI) bus, you can attain about 132 MBps. With the PCI-X bus and a SCSI controller, you could reach 450 MBps. (Source: *Samba-3 By Example*, John H. Terpstra). An alternative to local file storage is to move file storage to a Storage Area Network (SAN) or Network Attached Storage (NAS), offloading disk and network I/O to hardware designed specifically for this purpose. A SAN allows the administrator to invest money into one system rather than each DC.

If you don't have enough memory, the server will spend more time on seeks and more paging out to disk (slow), and operations will have to wait until resources are available, causing sluggish performance and possible locking up the server. If it can be budgeted, maximize the amount of memory in your servers from the beginning. I have seen the following per-user per-service memory estimates at: 2.5MB dynamic host control protocol (DHCP), 16MB Domain Name System (DNS), 16MB NMBD, 16MB WINBIND 4MB SMBD, 10MB APACH, 3.5MB cups, and 256MB base OS (Source: *Samba-3 By Example*, John H. Terpstra)

Migrating File Services to Linux

Migrating file services to Samba is a multi-step process. Unfortunately, there are no ready-made scripts to do this; however, there is work underway to add the necessary commands to Samba to obtain the information that would allow easier scripting of the process.

Before you begin to migrate shares and files to the Samba server, make sure it has properly joined the domain and that you have set up all of the required user accounts and any username mappings that are needed. If you are running as a Domain Member Server, make sure *winbindd* is operating properly to allow mapping of the Windows SID (Security Identifier) to a UNIX UID. If this is not working, you will not be able to copy the ACL settings on the files and you may even get permission errors when trying to create the files or directories.

First you must determine the names of the shares on the Windows server and create those shares and directories on the Samba server. You can use the *net rpc share* command to get a list of shares on a remote server. Let's assume the server is called *ntserver* and has an administrator password of *admin*. You could use the following command to list the existing shares:

```
net rpc share -S ntserver -Uadministrator%admin
```

Add the appropriate share definition entries to your *smb.conf* file and create the needed directories. You may need to restart Samba or issue a Hangup (HUP) signal to the SMBD processes to cause the new share to be visible to your Windows client machine.

There are two separate sets of permissions on shares in Windows that tend to confuse people. There are permissions applied to the share (share permissions) and permissions applied to files and directories. From a Windows 2000 client you can see the share permissions for folders it is sharing by right-clicking on a shared folder and clicking on **properties**. There is a tab for "Sharing" and one for "Security." On the Sharing tab there is a button for "Permissions" that brings up the "Share Permissions" tab that allows you to set the three share permission settings to allow or deny *full control*, *change*, and *read* permissions. The "Security" tab allows you to set permissions on the actual directory. From a Windows 2000 or later client, you can use *xcopy* with the */o* option to copy files to the Samba server and preserve the ACL settings (at least to the extent that Samba can translate these ACL entries to POSIX ACLs). You can also use the Samba utility called *smbcacls* to view and modify ACLs. If the machine you are migrating the shares from is not a Windows 2000 or later machine, you will have better luck (although it will be slower) mounting both the source and destination machines on a Windows 2000 machine and using *xcopy* from there.

The latest version of Samba (v.3.0.7) has problems setting the owners of files to be built-in groups (such as administrators); therefore, if you have files that are owned by groups (which is allowed in Windows) you will receive *access denied*

errors during the copy, and the owner and ACL entries will not be set properly. Hopefully, this will be fixed in the next release.

Migrating the File Permissions

While *xcaccls* may be your friend, it can't recurse directories on its own. The resource kit for W2k will install to *C:\Program Files\Resource Pro Kit\xcaccls.exe* and NT4 will install to *C:\Ntreskit\xcaccls.exe*.

Be sure to *add start->settings->control panel->system->Advanced->Environment Variables* to the PATH string for both "User variables for Registered User" and "System Variables." This command can be helpful for setting, viewing, and saving the permissions on all files and directories on an NTFS partition (it will not work on a FAT system). Use of the command is obscure, but here are a few useful usage examples:

To dump ACLs for a root file system:

```
xcaccls %systemroot%\*.* /T > C:\%1_acl.txt
```

To view permissions on a file:

```
xcaccls.exe c:\winnt
c:\WINNT BUILTIN\Users:R
    BUILTIN\Users:(OI)(CI)(IO)(special access:)
                                GENERIC_READ
                                GENERIC_EXECUTE

    BUILTIN\Power Users:C
    BUILTIN\Power Users:(OI)(CI)(IO)C
    BUILTIN\Administrators:F
    BUILTIN\Administrators:(OI)(CI)(IO)F
    NT AUTHORITY\SYSTEM:F
    NT AUTHORITY\SYSTEM:(OI)(CI)(IO)F
    BUILTIN\Administrators:F
    CREATOR OWNER:(OI)(CI)(IO)F
```

- **Inherit Only (IO)** ACE (Adaptive Communication Environment) does not affect this folder.
- **Container Inherit (CI)** ACE will be set on all folders in this directory.

- **Object Inherit (OI)** ACE will be set on all files in this directory.
- **Non-Propagate (NP)** ACE will not be applied to subfolders or files in this directory.
- **F** Full Control
- **C** Change
- **W** Write

Dump all permissions for files and directories on an NTFS file system to a file. According to Microsoft Knowledge Base Article 245015, “You can use the *Xcals.exe* utility to print the permissions for files and folders contained in a folder, but you cannot print the permissions for folders and files contained in the subfolders.”

```
XCACLS *.* > C:\filename.txt
```

There is freeware called FILEACL (www.gbordier.com/gbtools/fileacl.htm) that can be used to recursively dump all ACLs for local and remote shares on a network.

```
setup samba as bdc with replication
setup samba to respect linux acl

dumping NTFS acl's,

apply acl's to the shares.

drag and drop files to samba share
```

You can get a list of shares and share permissions on a windows system with the Resource Kit utility *srvcheck* (i.e., *srvcheck \hostname*) and you can look at file permissions with the Resource Kit utility *perms* (i.e., *perms C:\my_directory*).

Support Tools Utility showwacccs

The *SubInAcl* tool is a clever and effective tool for modifying share permissions.

```
save settings:      SubInAcl /output=c:\subinacl_save.txt /noverbose
/display
```

```
replay settings:   SubInAcl /playfile c:\subinacl_save.txt
```

```
/subdirectories
```

```
SubInAcl /subdirec \\server\share\*. * /display /noverbose
```

Modify share permissions from command line:

Good list of comamnd line tools:

<http://www.ultratech-llc.com/KB/ASP/FileView.asp?File=/KB/Perms.TXT>

To set up shares and permissions, you must determine the effective permissions for all users on all shares. The process for doing this is:

1. Check for a “NO ACCESS” trump.
2. Check most permissive SHARE permissions (they are a member of all groups).
3. Check most permissive FILE permissions (they are a member of all groups).
4. Effective permission over a network will be the more restrictive of the FILE and SHARE permissions.

TIP

If User A creates an Excel file F that User B will edit, when User B opens the file, Excel will create a new temporary file (owned by User B). Upon saving the changes, Excel will rename the temporary file to the original name, so that now user B is now the creator. User A would still own the file with a W2K server, .

Solutions Fast Track

Understanding Windows File Systems

- ☑ While Linux does have read/write access to FAT file systems, FAT/VFAT should be avoided as they lack ACL's, are inefficient with hard drive space, and fragment easily.
- ☑ The NTFS file system handles much larger partitions, supports RAID, complex ACL's, and supports journal logs for transactions, and the features keep growing. The drawback is that compatibility with Linux is limited to a read-only kernel driver (with the exception of the Captive project), making it a non-option for non-NT based Microsoft Operating Systems.
- ☑ The typical Windows file systems (FAT/VFAT/NTFS) are limited (intentionally or not). There are more robust and feature-filled options available to Linux and other *nix operating systems, so be prepared for the transition when moving to Linux.

Understanding Linux File Systems

- ☑ EXT2, EXT3, and ReiserFS now support POSIX ACL's, RAID, and read/write access is available to both Linux and Windows Operating Systems. They are the dominant filesystems used by Linux. One can convert an ext2 filesystem to ext3 and vic-versa.
- ☑ Ext3 and ReiserFS journal, with different levels of journaling, offer a more dependable filesystem to the enterprise. ReiserFs now claims to be the fastest available filesystem.

Understanding Permissions Management (Access Control)

- ☑ You can get Samba and Linux file systems to support POSIX ACL's, for more granular permissions, similar to those supported by NTFS, but adds unnecessary complexity to preserving the ACL's with a backup solution or when moving data from one server to another.

- ☑ The long-term goal should be to attain a sufficient level of granular access control by using standard Unix file permission with Samba share permissions.

Understanding File Backup, Restore, and Replication Options

- ☑ Files and servers will encounter damage or loss, making it necessary to engineer fileservices with an integrated backup and restore plan.
- ☑ Back up only non-standard files (to minimize backup time and resources), and organize files to be backed up together.
- ☑ Choose a back-up method and media that is appropriate for the size and complexity of your network. This will aid your back up schedule window.
- ☑ Schedule regular back-ups, full or incremental, to match your media and time resources constraints, to also ensure one can perform a restore within the expected amount of time.
- ☑ Carefully plan to archive, reuse and discard media as near-line and off-line storage, to ensure a level of dependable media and long-term access to backups while avoiding the need to archive all back ups.

AMANDA

- ☑ AMANDA is an extensible Open Source client/server backup solution for network backups of *nix and Windows clients, which runs on *nix, with a command line interface, using standard *nix utilities including rsync, dump, tar, star, or cpio, and mt.
- ☑ The backp process starts when server contacts the client daemon. The client collects the data and data is sent back to the server. The server stores the data in a holdingdisk area, before writing the storage media.
- ☑ With a few configuration files, you can create a dependable network backup solution free of commercial licenses.

Designing Linux-based File Services

- ☑ On the network, set at least 2 domain controllers (with WINS) per network segment.
- ☑ With hardware, plan on a data transfer of 100Kbit/sec per user, from network-to-memory-to-disk and back.
- ☑ Symplify the network from the client's perspective, and ease the WAN traffic by using rsync between servers on network segments. Set up DFS for each of the shares. Version control will be needed for files that are modified on different network segments, to handle simultaneous changes.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the "Ask the Author" form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Can Samba replace a Windows NT4 Domain Controller or Active Directory server?

A: Samba-3 can replace a Windows NT4 Domain Controller, but cannot replace an Active Directory server yet.

Q: What are some simple troubleshooting solutions to common problems?

A: These are some troubleshooting steps for setting up a Samba server:

'testparm' finds no errors in smb.conf

'lsf -i tcp ; lsf -i udp' show running daemons and ports: ldap, nmb, smb, windbind, cups.

'smbclient -L localhost -U%' shows samba is responding, host MYSERVER is master for MYDOMAIN.

'net getlocalsid' and 'net getlocalsid MYDOMAIN' shows the same SID, so MYSERVER is in the MYDOMAIN.

'slapcat' dumps all the objects in the ldap database (even when ldap is not running).

'ldapsearch -x -b "dc=myscompany,dc=com""(ObjectClass=*)"' proves slapd is running and responds.

'getent passwd |grep Administrator' shows that nsswitch.conf is configured to send me to the LDAP server using the nss_ldap library.

'pdbedit -Lv Administrator' proves that samba can get info from the ldap backend.

'getent group' shows all the standard domain groups (as well as unix groups).

Print Services

Solutions in this Chapter:

- Understanding Windows Print Services
 - Understanding Linux Print Services
 - Sharing Samba Printers
 - Understanding Automatic Printer Driver Downloading
 - Migrating Windows Print Services to CUPS/Samba
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Printing is one area that causes a great deal of confusion among users. This is mainly because it involves setup issues on both the client and server machines. If either side is configured improperly, nothing you do with Samba (which only serves to connect the two halves) will be able to make printing work. With this in mind, this chapter will walk through the printer setup steps on both the server and client machines as well as the Samba-specific steps required to make the two sides communicate properly.

This chapter assumes that you understand basic administrative tasks on a Windows client such as bringing up the Add Printer Wizard or locating a printer in the Network Neighborhood. It is also assumed that you have already installed and configured CUPS on the Linux system. Only brief instructions will be given for these tasks.

An overview of the Windows and Linux printing services will be given. This will be followed by step-by-step instructions for configuring a printer for best use with Samba and Windows clients. Finally, the steps necessary to configure the printer on the Windows client to allow for *point and print* will be described as well as troubleshooting steps to try when things just don't seem to be working as you expected.

Understanding Windows Print Services

The Windows printing subsystem uses the Graphics Device Interface (GDI). Objects that are displayed on your screen and things sent to your printer use the same library of routines to convert the output to a standard language known as Enhanced Meta File (EMF). This is then processed by the actual display or printer driver. This makes WYSIWYG (What You See is What You Get) printing much simpler.

When a Windows print client connects to a Windows print server, it can automatically receive and install the proper print driver for the printer. This is known as *point and print*. The server can store drivers for various versions of Windows for each printer attached so that clients running versions of Windows that are different from the version of Windows running on the server can receive the proper driver for the selected printer. When this driver is installed on the client, it can run code to update registry settings to configure the default printing options to be used for the printer. For PostScript printers, the printer driver basically consists of a file known as a PostScript Printer Description (PPD)

file (provided by the printer manufacturer) that maps all the features supported by the printer to appropriate PostScript, PCL or Printer Job Language (PCL) commands understood by the printer. The printer driver GUI uses this file to present the user with the proper user-selectable print options for the printer.

In a purely Windows environment, when an application attempts to print, the following process occurs:

1. The application will make calls to the GDI to convert the output to EMF format.
2. The printer driver is used to convert the EMF output to a format that can be understood by the printer.
3. The printer-ready file is spooled on the print server machine.
4. The file is sent to the printer.

Which of the steps are executed on the client and which are executed on the server can vary. The client can choose to convert the EMF data itself using the printer driver locally or it can send the EMF data to the server and have the server run the printer driver.

The print server can set up access restrictions for each printer that can limit which clients can connect, restrict the hours that the printer can be used, or restrict which users can view and control printers and print jobs. The client or server can use the print queue application (invoked by clicking on the printer icon) to view the status of print requests, pause or resume printing of a request, or cancel a request if the logged-on user has been granted permission.

Understanding Linux Print Services

UNIX systems didn't develop a common interface for display and printing like the Windows GDI. Consequently, each application formats the printer output however it chooses. Image files tend to be in Graphics Interchange Format (GIF) or Joint Photographic Experts Group (JPEG) format and most other types of files are output in PostScript. Many of the printers used on UNIX systems actually contain a PostScript interpreter built into the printer that converts the PostScript input to the raster image required by the print engine. For other types of printers, the filter programs on the print server take care of converting the PostScript or image files to a format suitable for the printer.

In simple terms, printing can be thought of as a three-step process.

1. The file is stored on the print server and placed into a queue until the printer is ready to print. This step is called *spooling* the print file.
2. The file may have to be translated into a form that the printer recognizes. This step is called *filtering* the print file.
3. The file (in the proper format for the printer) is sent to the printer where it is printed. This printer can be locally attached to the computer (serial, parallel, USB, etc.) or accessed over the network. The spooling system also needs simple commands to allow someone to list the status of printers and pending print jobs, pause or resume printing of a job, and a means to cancel printing of a pending job.

Printing is one area where UNIX systems have diverged more than other areas. There are two basic camps of UNIX systems, one that is System V (SYSV)-based and one that is Berkeley (BSD)-based. Each of these two UNIX types uses a different spooling method and a different set of commands to manipulate the print spool. Many versions of UNIX today will supply both sets of commands allowing the user to choose the most familiar style. Adding to this confusion, not all UNIX vendors will use the same command line switches to the spooling programs. New and improved versions of the print spoolers have been introduced to add even more confusion. The new features added by the Common UNIX Print System (CUPS) printing system will be discussed in more detail later.

Table 6.1 lists the printing commands that are typically found on a SYSV-style printing system. For more details and a description of the options available for each command, check the man page for the command.

Table 6.1 System V Printing Commands

SYSV Command	Description
lpsched	Daemon that schedules print jobs (normally started by the system at boot time)
lpshut	Turns off the print scheduler daemon
enable	Enables a printer for use by the scheduler and allows jobs to be printed
disable	Disables printer from use by the scheduler and prevents jobs from being printed
reject	Prevents lp from accepting jobs for a specified printer
accept	Allows lp to accept jobs for a specified printer

Continued

Table 6.1 System V Printing Commands

SYSV Command	Description
lpmove	Moves printer jobs from one printer to another
lp	Sends a print job to a printer's queue
cancel	Cancels a print job
lpstat	Displays printer and print job status information

Table 6.2 lists the printing commands that are typically found on a BSD-style printing system. For more details and a description of the options available for each command, check the man page for that command.

Table 6.2 BSD Printing Commands

BSD Command	Description
lpd	Daemon that schedules print jobs (normally started by the system at boot time)
lpc	Printer control program used to enable/disable a printer, rearrange jobs in the queue, and determine status of printers and their queues
lpr	Send a print job to a printer's queue
lprm	Removes a print job from the print queue
lpq	Displays printer and print job status information

Configuring Linux Printing Using BSD or SYSV

This section will discuss the steps necessary to enable your UNIX system to print to a printer that is directly attached to the system. The details are different depending on whether your printing system is BSD- or SYSV-based, but basically you need to create the appropriate directories and printer filter files and inform the system about the location and capabilities of the printer. For the following examples we will assume your printer has been named *myprinter*.

On SYSV-based systems you should create your model interface file or use one of the model files from the system, and use **lpadmin** to install that printer. This will create the necessary spool directory, add the interface file as `/var/spool/lp/interface/myprinter` and create the file `/var/spool/lp/member/myprinter` indicating the physical device where the printer is attached. You then run **enable**

myprinter followed by **accept myprinter** to enable the printer and allow jobs to be spooled.

On BSD-based systems, you need to create the **/etc/printcap** entry and any filter programs necessary, create the spool directory (usually */var/spool/lpd/myprinter*), and then use **lpc** to enable the printer and its queue.

Most versions of UNIX have a GUI tool or scripts for adding printers to the system. It is recommended that you use one of these tools to install your printer unless you are an administrator with extensive experience in installing printers. Creating **/etc/printcap** entries or printer filters can be somewhat tricky and teaching this is beyond the scope of this book.

Once you have installed your printer on your UNIX system you are now ready to print a document. For these examples we will assume your printer has been named *myprinter* and you have a text file you wish to print that is called */tmp/myfile*.

In order to print on a BSD-style system you would issue the command:

```
lpr -Pmyprinter /tmp/myfile
```

This command causes the **lpr** program to copy the file */tmp/myfile* to the spool directory */var/spool/lpd/myprinter*. The print spooler daemon program (**lpd**) then consults the printer capability database */etc/printcap* to determine how to process the file and where to send the output.

In order to print on a SYSV-style system you would issue the command:

```
lp -dmyprinter /tmp/myfile
```

This command causes the **lp** program to make a link to the file */tmp/myfile* in the spool directory */var/spool/lp/request/myprinter* (if you wish to make a copy instead of a link you can add the **-c** option to the command above). The print spooler daemon program (**lp sched**) then processes the file and sends it to the proper place based on the information found in the interface program found in */var/spool/lp/interface/myprinter*. If you do not specify the **-c** option to create a copy of the file, you must insure that the original file is not deleted before the printing is completed.

You must ensure that you can print a file on your UNIX system using one of the above commands before you attempt to make this printer available through Samba. Most printers on UNIX systems can print text files (also known as ASCII files) or files in the PostScript page description language (files that start with the characters %!).

Once you are able to print correctly, you need to ensure that the user you have selected as the guest account user for Samba is also able to print correctly. Since your guest account user does not necessarily have the ability to log into your UNIX machine, you can check the ability to print from this account by first becoming root on your machine and using the **su** command to then become the Samba guest account user.

Sometimes the printing commands may not be in the search path for your user. You can use the **whereis** command to determine the full pathname of the command. Assuming your guest account user for Samba is named **nobody** you would execute the following commands on a BSD-style UNIX:

```
[root@linuxbox /]# su - nobody
[nobody@linuxbox /]$ whereis lpr
lpr: /usr/bin/lpr /opt/./bin/lpr /usr/man/man1/lpr.1
[nobody@linuxbox /]$ /usr/bin/lpr -Pmyprinter /tmp/myfile
```

This tells you that the **lpr** command is found at **/usr/bin/lpr** on this system. You can then use the full path to execute the command to print your file.

Configuring Linux Printing Using CUPS

Unlike the traditional BSD or SYSV printing described above, the Common UNIX Print System is much more than just a print spooling system. It is a complete printer management system based on the Internet Printing Protocol (IPP), which allows local or remote management of your print server. Many of the management functions can even be accessed via a web browser to give you a platform-independent method for managing all your printer services. It also contains the traditional BSD and SYSV command line interfaces as well as several third-party GUI interfaces. The current version of CUPS uses Ghostscript to convert image files and PostScript files to the raster images required for non-PostScript printers. You can run the command **ghostscript -h** to get a list of the devices supported by your version of Ghostscript.

As you may recall, in the Windows printing model, the server can receive a job in either a completely printer-ready format or in a format that needs conversion (EMF). The CUPS system supports creating raw printers where no output file translation is required or smart printers where CUPS does the translation to a format understood by the printer. CUPS also makes use of the PostScript Printer Description file to describe the capabilities of the printer. CUPS extends this notion to non-PostScript printers as well and many PPD files are available for non-PostScript printers. By having PPD files for all printer types, CUPS is

able to have all printers appear as PostScript printers to the clients and allow the client to be able to select all the supported printer options. CUPS has developed its own PostScript driver for Windows NT/200x/XP, or you can use the Adobe PostScript driver for earlier versions of Windows.

A complete explanation of how CUPS works is beyond the scope of this chapter and there are many sources of information on the detailed workings of CUPS available. After installing CUPS, all of the documentation will be available to you. The Samba HOWTO collection documentation contains a very good description of how CUPS works and how to use CUPS. Basically, CUPS sets up filter chains for incoming files to convert them to PostScript and then processes this PostScript using the external GhostScript program to the proper printer format. This is then transferred using the CUPS backends to the printer. The printer may be locally attached (parallel, serial, etc.) or accessed through a TCP/IP network (IPP, LPR/LPD, SMB, etc.).

By having a common input format (PostScript), CUPS can easily perform such functions as page counting, accounting, printing multiple pages per sheet (n-up printing), etc. Since CUPS is a complete printer management system and not just a spooler, it also has provisions for setting up quotas to limit the number of pages or size of jobs that can be submitted by users, access controls, authentication, and even encryption. You can even set up several printers into a printer class for highly available printing.

NOTE

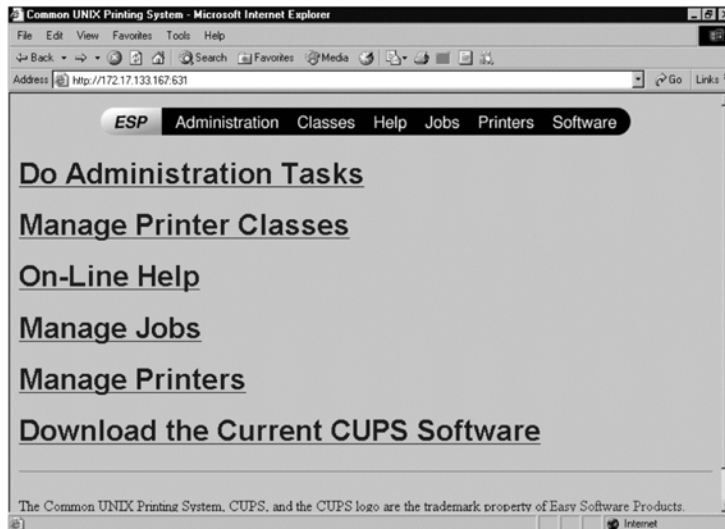
If you set up a printer as a raw printer (the input is sent directly to the printer untouched) then the accounting functions will be bypassed and the job will only be counted as a single page.

You can add and manage your printers using the standard BSD or SYSV commands described above, one of several GUIs such as **kdeprint**, or you can use the web interface that installs with CUPS. Once you have installed and started CUPS, bring up your browser and connect to <http://localhost:631> to access the online documentation or access all the management and administration functions, as shown in Figure 6.1.

NOTE

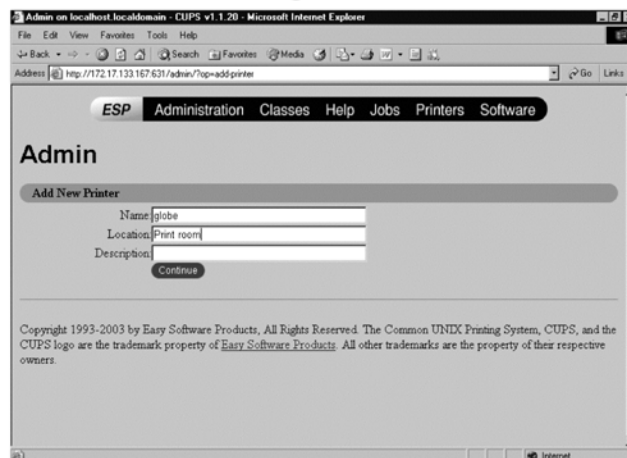
By default, CUPS is configured to only allow access from localhost, but it can be configured to allow connections from other machines as well.

Figure 6.1 CUPS Web Interface Start Page



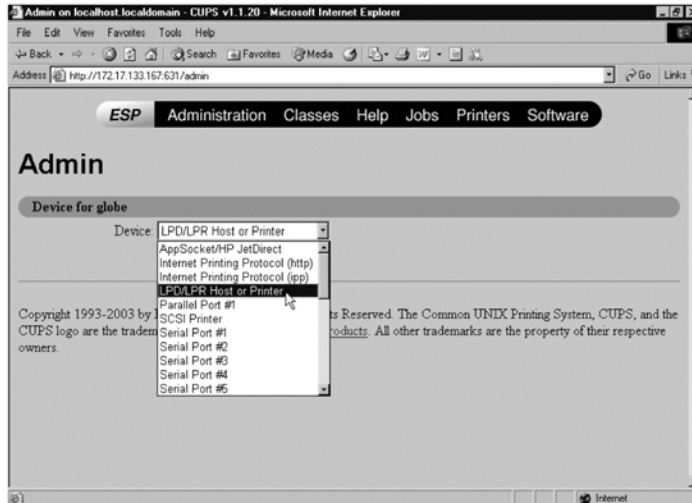
1. If you select **Administration** you will be prompted for a username and password.
2. Enter **root** for the username and enter the proper password. We'll walk through setting up an HP LaserJet printer named **globe** on a remote **LPR/LPD** server named **server.mycompany.com**.
3. From the Admin screen, select **Add Printer**. You will be presented with the screen in Figure 6.2 that allows you to enter a name, location, and description for this printer (only the name is required).

Figure 6.2 CUPS Add Printer Page



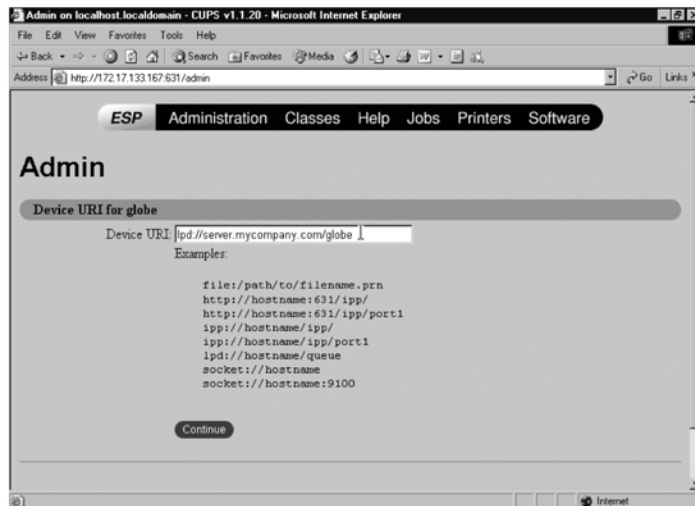
- When you click **Continue** you will be presented with a screen to select the device for the printer. CUPS supports printers connected locally to parallel, SCSI, serial, or USB ports and network printers using AppSocket (HP JetDirect), HTTP, IPP, or LPD/LPR. SMB-connected printers are also supported via Samba. For this example we will use LPD/LPR, as shown in Figure 6.3.

Figure 6.3 Configuring a Printer Device



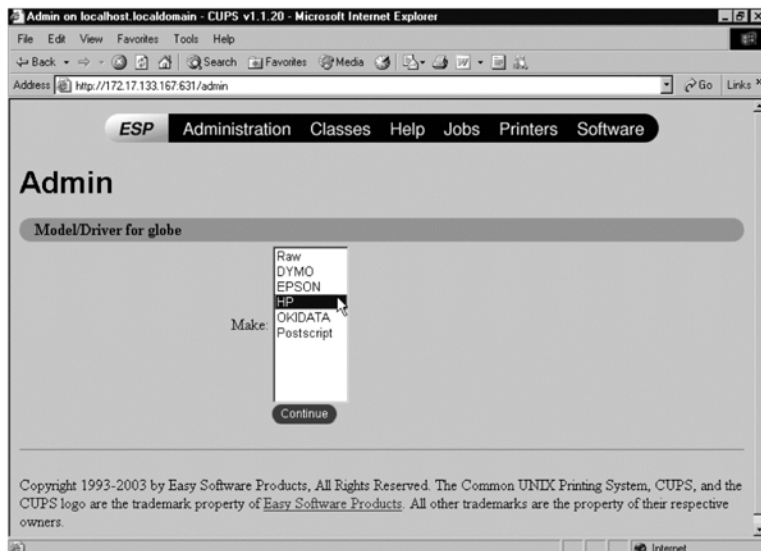
- When you click **Continue** you will be presented with a screen (shown in Figure 6.4) to select the Device **URI** (Uniform Resource Identifier) for the printer. These are formatted similarly to the URL (Uniform Resource Locator) used in web browsers. Examples for the various types are given on the screen. For this example, we are using **LPD** protocol to a host named **server.mycompany.com** that has the printer queue name **globe**.

Figure 6.4 Specifying the Device URI



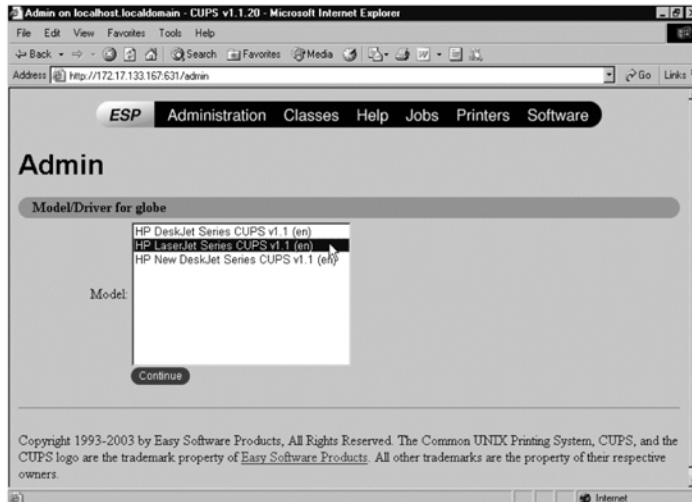
6. Enter the URI and click on **Continue**. You will now be able to select the **Make** of your printer (shown in Figure 6.5). CUPS includes drivers for several printer makes, and many more are available in the commercial version of CUPS available from Easy Software Products at www.easysw.com/printpro. There are additional drivers available at <http://gimp-print.sourceforge.net> and www.linuxprinting.org.

Figure 6.5 Specifying the Printer Make



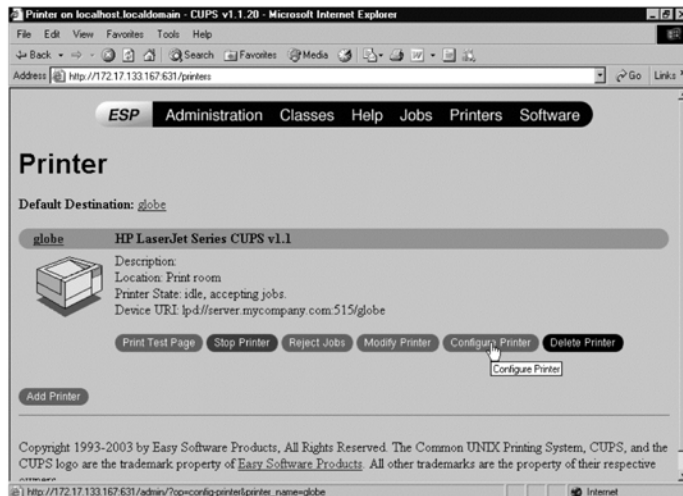
7. Select **HP** for our example and click **Continue**. You will now be able to select your printer **Model** on the next screen, shown in Figure 6.6. For our example we will select the **LaserJet Series**.

Figure 6.6 Specifying the Printer Model



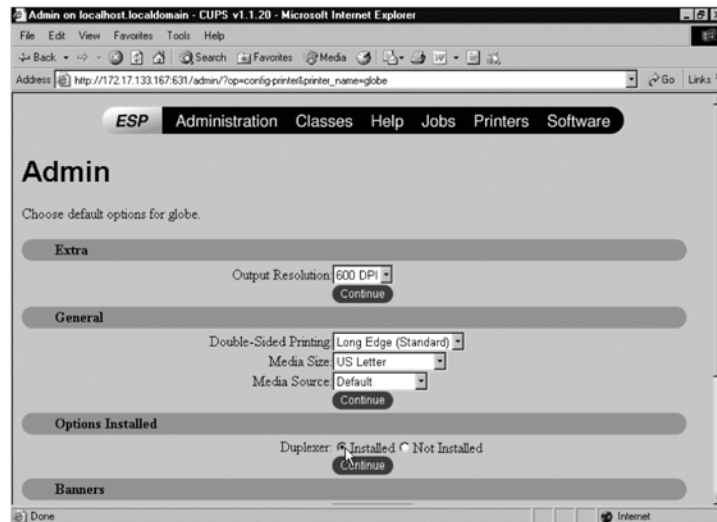
8. When you click **Continue** you will have completed setting up your printer. If you now click on the **Printers** button at the top of the page you will see the information for the printers you have on the system as well as buttons for various tasks, as shown in Figure 6.7.

Figure 6.7 Viewing the Printer Settings



9. You will need to configure the options for the printer. Click on the **Configure Printer** button and you will be presented with a screen (shown in Figure 6.8) that allows you to select values for all the options specified in the PPD file for this printer. After selecting all the proper values click on any of the **Continue** buttons.

Figure 6.8 Configuring Printer Settings



10. You can now test the printer by clicking on the **Printers** button at the top of the page and then clicking on **Print Test Page**. Make sure this works before trying to configure this printer to work with Samba. Test printing as the user specified by Samba in the same manner as described in the previous section.

Sharing Samba Printers

Once you have your printer working from your UNIX system you can now configure Samba to make that printer available to Windows clients. Before setting up the actual share, let's review the various *smb.conf* parameters that especially pertain to printing. These are the parameters that you are able to view and change through the Samba Web Administration Tool (SWAT). Table 6.3 lists printing parameters found in **smb.conf**. Some parameters have aliases that are indicated in parentheses after the primary name.

Table 6.3 Samba smb.conf Printing Parameters

Parameter Name	Description
printcap name (printcap)	A global parameter that is used to override the compiled-in default printcap name used by the server to obtain the list of printers.
load printers	A global parameter that indicates if all printers from printcap will be loaded for browsing by default.
printable (print ok) Printing	Allows spooling of print files to the service. Sets the printing style for your system, which determines how printer status information is interpreted. This parameter controls the default values of print command, lpq command, lppause command, lpresume command, and lprm command.
print command	After a job has finished spooling to a service, this command will be executed via a <code>system()</code> call to process the file.
lpq command	The command used to obtain printer queue status information.
lprm command	The command used to delete a specific print job.
lppause command	The command used to stop printing or spooling a specific print job.
lpresume command	The command used to restart or continue printing or spooling a specific print job.
queuepause command	The command used to stop print jobs in the printer queue from being submitted to the printer.
queueresume command	The command used to resume sending jobs in the printer queue to the printer
Comment	A text field that appears for a share when a client does a query to the server for a list of available shares.
path (directory)	The directory where print data will spool prior to executing the command in print command.
Printer (printer name)	The name of the printer where print jobs will be sent.

Continued

Table 6.3 Samba smb.conf Printing Parameters

Parameter Name	Description
Postscript	This forces a printer to interpret the print file as PostScript by adding a %! to the start of print output.
lpq cache time	A global parameter that controls how long information returned by the command specified in the lpq command will be cached to prevent this command from being called too often.
min print space	This sets the minimum amount of free disk space that must be available before a print job can be spooled.
guest account	This is the UNIX username that will be used for access to services that are specified as guest ok.
guest ok (public)	Specifies that no password is required for this service. Access will be as the user specified in guest account.
hosts allow (allow hosts)	A list of machines that will be allowed access to this service. May be specified by name or IP address.
hosts deny (deny hosts)	A list of machines that will specifically NOT be allowed access to this service. May be specified by name or IP address. If there are conflicts, the hosts allow list takes precedence.
browseable (browsable)	This controls whether this service will show up in client browse lists.
Available	This allows a service to be disabled. All attempts to connect will fail.
Preexec (exec)	This specifies a command to be executed at each connection to a service.
root preexec	This specifies a command to be executed as root at each connection to a service.
Postexec	This specifies a command to be executed whenever the service is disconnected.
root postexec	This specifies a command to be executed as root whenever the service is disconnected.

In addition to the parameters listed above, most service-level parameters can be included in a printer service. There are seven variables specifically for printing

and several others that are useful in creating the various commands described above. Table 6.4 summarizes these variables.

Table 6.4 smb.conf Printing Variables

Variable	Definition
%f or %s	The file name only (without the path) of the file to be printed.
%p	The name of the UNIX printer to use.
%j	The number of the print job (for use in the print queue control commands).
%J	The job name as sent by the client.
%c	The number of printed pages (if known).
%z	The size of the spooled file (if known).
%S	The name of the current service.
%U	The session user name (the username the client desired, not necessarily the same as the one they got).
%m	The NetBIOS name of the client machine.
%L	The NetBIOS name of the server.
%T	The current date and time.

Before setting up the printer share in Samba you need to determine what printing style your system is using. The Samba binaries will have a default printing style that was set at compile time. The default values of **printcap name** and guest account are also set at compile time. You can determine the default settings for your binaries by creating an empty configuration file (or using */dev/null*) and running **testparm** against that file. To find the default **printing** style, **printcap name**, and **guest account**, run the following commands:

```
[root@linuxbox /]# testparm -sv /dev/null | grep printing
printing = bsd
[root@linuxbox /]# testparm -sv /dev/null | grep printcap
printcap name = /etc/printcap
[root@linuxbox /]# testparm -sv /dev/null | grep guest account
guest account = nobody
```

If the default is incorrect, you can change it by adding a **printing = <desired value>** entry and a **printcap name = <path to printcap>** entry to the **global** section of your **smb.conf** file. It is a good habit to place these entries

in your **smb.conf** file even if the defaults are correct. This makes it perfectly clear what values will be used. The printing style then determines the defaults for the other printing commands. Table 6.5 shows the default printing commands that are set for each printing style. If these commands are incorrect for your system, you may change them in the global section of the **smb.conf** file.

Samba currently supports the following printing systems: SYSV, AIX, HP-UX, BSD, QNX, PLP, LPRNG, CUPS, NT, and OS2. Each of the last two types (NT and OS2) is LPR on the respective OS. The entries for CUPS are the values you would get if Samba was not compiled with CUPS support (does not use libcups). If Samba was linked with libcups, it will use the CUPS API to send print jobs and control the printer and print queue and all entries for the default commands will be ignored. If you need to set a specific print command or one of the other commands when you are using CUPS printing with Samba using libcups, you can set the printing style to **printing = sysv** in the specific printer section and override the command there. This allows CUPS to control all the other printers and only override the settings for one specific printer.

Table 6.5 Printing Style Defaults for Various Printing Systems

Printing Style	Default commands
BSD,AIX,NT,OS2	print command = lpr -r -P'%p' %s lpq command = lpq -P'%p' lprm command = lprm -P'%p' %j lppause command = lpresume command = queuepause command = queueresume command =
LPRNG, PLP	print command = lpr -r -P'%p' %s lpq command = lpq -P'%p' lprm command = lprm -P'%p' %j lppause command = lpc hold '%p' %j lpresume command = lpc release '%p' %j queuepause command = lpc stop '%p' queueresume command = lpc start '%p'

Continued

Table 6.5 Printing Style Defaults for Various Printing Systems

Printing Style	Default commands
CUPS	print command = /usr/bin/lp -d '%p' %s; rm %s lpq command = /usr/bin/lpstat -o '%p' lprm command = /usr/bin/cancel '%p-%j' lppause command = lp -l '%p-%j' -H hold lpresume command = lp -l '%p-%j' -H resume queuepause command = /usr/bin/disable '%p' queueresume command = /usr/bin/enable '%p' printcap name = lpstat
SYSV	print command = lp -c -d%p %s; rm %s lpq command = lpstat -o%p lprm command = cancel %p-%j lppause command = lp -i %p-%j -H hold lpresume command = lp -i %p-%j -H resume queuepause command = disable %p queueresume command = enable %p
HPUX	print command = lp -c -d%p %s; rm %s lpq command = lpstat -o%p lprm command = cancel %p-%j lppause command = lpresume command = queuepause command = disable %p queueresume command = enable %p
QNX	print command = lp -r -P%p %s lpq command = lpq -P%p lprm command = lprm -P%p %j lppause command = lpresume command = queuepause command = queueresume command =

Next we create the necessary entries in **smb.conf** to share a printer. The printing style will be **CUPS**, the printer name on the UNIX system will be **lp** and we will share the printer under the name **myprinter** and allow everyone to have access to the printer. The files will be placed in a directory called **/var/spool/samba** with permissions **0700** before they are sent to the system print spooler. No other printers on the system will be made available for sharing at this time. The following entries show what is needed in **smb.conf**.

```
[global]
printing = cups
printcap name = cups
guest account = nobody
load printers = no

[myprinter]
printable = yes
writeable = no
path = /var/spool/samba
guest ok = yes
printer = lp
create mask = 0700
browseable = yes
```

The **printing = cups** entry in the global section sets the other printing commands to the values listed in the table above or use the CUPS API if Samba was compiled to use libcups. The **load printers = no** entry specifies that all the printers managed by CUPS will not all be shared automatically. The share **myprinter** is then specified and declared as a printer with the **printable = yes** entry. Printer shares will always allow writing to the directory specified in the **path =** parameter (UNIX user privileges permitting) via the spooling of print data. The **writeable = no** entry specifies that non-printing access to the share will not allow writing. Specifying **ok = guest** sets the share to allow all users access as the user named in the **guest account** parameter with no password required.

Because the printer name on UNIX is not the name we wanted to appear to Windows clients, we specify the UNIX name with the **printer = lp** parameter. If this parameter is not specified, it is assumed that the UNIX printer name will be the same as the share name. It is important to remember here that even though the share name is not case-sensitive to the Windows clients, the name specified in the **printer =** parameter (or the share name if this parameter is not specified) must exactly match the printer name as it is defined on your system.

The **create mask = 0700** entry causes all group and other permissions to be stripped from the file that is spooled to the share.

Finally, the **browseable = yes** entry allows this printer to show up to SMB clients when they browse the server from the Add Printer dialog box. You can set this to **no** if you want this printer to remain hidden to clients. They will still be able to connect to the printer if they supply the correct name.

If you only have a single printer attached to your system, it is fairly easy to create the share definition in your **smb.conf** file. If you have several printers, Samba has a mechanism called the **printers** section that allows you to easily share all printers attached to your system by creating only one section entry in your **smb.conf**. The limitations of using this method are that all printers must have the same parameters (with the exception of the comment) and the printer name must be the same as the share name.

Let's assume we have three printers on our system called **hp**, **slides**, and **color**. We could create one section in the **smb.conf** file that would share all these printers with the same parameters. The following entries show what should be added to the **smb.conf** file.

```
[global]
printing = cups
printcap name = cups
guest account = nobody
load printers = yes

[printers]
printable = yes
writeable = no
path = /var/spool/samba
guest ok = yes
create mask = 0700
browseable = no
```

NOTE

Do not set the path where files will be spooled to the same directory as your system print spooler uses or you may see unpredictable results when printing with Samba.

Let's take a look at what has changed from the previous example. In the **global** section we changed the **load printers** parameter to be **yes**. This is used in conjunction with the **printers** section. When a client attempts to connect to a share on the server, Samba will first check all the defined sections in **smb.conf**

for a match. If none is found, Samba will then see if a **homes** section is defined and if so, will attempt to match a user name in the local password file. If there is still no match and the **printers** section exists, Samba will scan for a match in the location specified by **printcap name**. Note that on a SYSV system this could be specified as the **lpstat** command, which will list the available printers.

If a match is found, a new section is created by copying the information in the **printers** section with the share name set to the located printer name. If the **printer** parameter is not present, the name of the printer is set to be the same as the share name. If guest access is not allowed and no username was given in the connection request, the username is also set to the name of the located printer. If you do not include the **printable** parameter, you will receive a warning message in your **smbd** log file and Samba will set the section to printable for you. The **browseable** parameter specifies whether the share name printers will appear in the browse list of printers. Since this share name is not actually a printer name, it is a good idea to set **browseable = no** to reduce confusion.

You can still use the **printers** section even if you do not want to share all printers attached to your system. You can accomplish this by specifying a pseudo-printcap that contains only the names of the printers you wish to share. Each line of this file consists of a printer name and any possible aliases in the following format:

name | alias | alias | alias with spaces

The alias names are not required. If the alias contains spaces, this alias is substituted for the comment parameter in the created share.

The **printers** section can also be used as a template for creating a printer share that will have slightly different parameters than other printers. Maybe you would like to limit which people can use the color printer defined above. You might add the following section to your **smb.conf** file.

```
[color]
copy = printers
guest ok = no
browseable = yes
valid users = @sales
```

This would cause the printer share **color** to be defined with all the parameters contained in **printers** and then modified to disallow guest access, make the share browseable, and only allow users in the sales group to have access. By using the **copy** parameter, you do not have to include all the common parameters in

the new definition so parameters such as **printable**, **path**, etc. do not have to be specified in the definition.

Understanding Automatic Printer Driver Downloading

Samba can be configured to automatically download printer drivers to Windows clients just as a Windows NT (or later) print server would do. This involves creating a new share called **print\$** where the driver files will be located, installing the driver files to the proper location in the **print\$** share, binding the driver to a printer share, and setting up the default printer options.

Creating the **print\$** share involves creating the proper directories on the Samba server and setting the proper **smb.conf** file entries. To support downloading drivers to various versions of Windows, you will need to create various subdirectories under the directory you created for the **print\$** share. Table 6.6 lists the subdirectory required for each architecture you might wish to support.

Table 6.6 Printer Driver Architecture Subdirectories

Subdirectory Name	Driver Architecture
W32X86	Windows NT x86
WIN40	Windows 95/98
W32ALPHA	Windows NT Alpha_AXP
W32MIPS	Windows NT R4000
W32PPC	Windows NT PowerPC

You can create a group on your Samba server to specify which users will have printer administrator rights (the ability to add new drivers and set printer properties). Assuming the group name you created is called **ntadmin**, you can add the following entry to the **global** section of the **smb.conf** file.

```
printer admin = @ntadmin
```

The **print\$** share entry might look as follows:

```
[print$]
comment = Download area for printer drivers
path = /var/samba/drivers
guest ok = yes
```

```

browseable = no
read only = yes
write list = @ntadmin, root

```

If you only want authenticated users to be able to access the printers, you can remove the **guest ok = yes** parameter. By setting **browseable = no**, you prevent the share from showing up in Network Neighborhood (by default, shares ending with a **\$** will not show up for anyone but an administrator anyway). Setting **read only = yes** prevents anyone from uploading driver files to this share by denying them write permission. The **write list = @ntadmin, root** entry allows root and people in the UNIX ntadmin group write permission so they can upload new drivers. Make sure the permissions on the share directories also allow write permission for the **ntadmin** group, as Samba will not override the underlying file system permissions.

The next step is to install the drivers for the printer onto the Samba server into the proper directory of the **print\$** share. This can be done using the Samba **rpcclient** command or using Windows Add Printer Wizard from an NT/2000/XP client. If you wish to use **rpcclient**, check the Samba documentation for instructions on how to do this, as this can become quite involved and will not be described here.

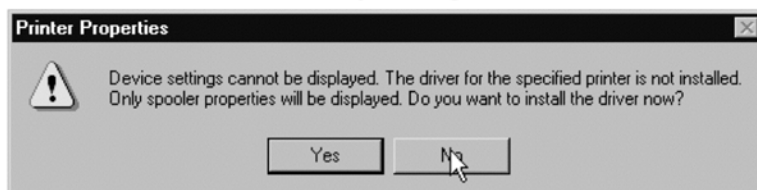
Using Windows Explorer, browse to the Network Neighborhood and open the Samba host. You will see the **Printers** folder and when you open this you will see all the printers that are being shared by the Samba host. Figure 6.9 illustrates right-clicking on the printer icon and selecting **Properties**.

Figure 6.9 Displaying Properties for a Samba Printer



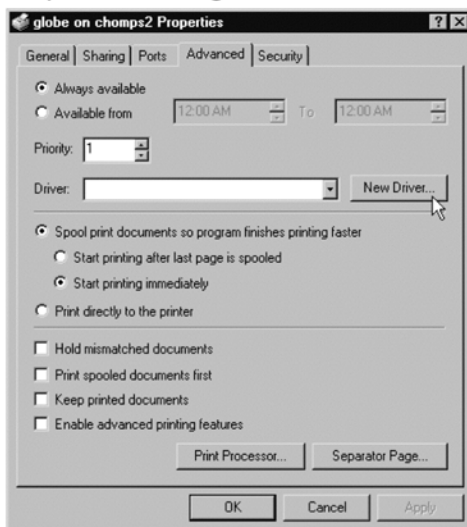
Since there is no driver associated with this printer, you will get an error message similar to the message shown in Figure 6.10. The wording will vary depending on which version of Windows is running.

Figure 6.10 Driver Installation Query Dialog Box



Do not click on **Yes**, as that will install the drivers on the local system. When you select **No** you will be presented with the **Printer Properties** dialog box. Select the **Advanced** tab, which will now allow you to select **New Driver** (see Figure 6.11) and install the driver to the Samba server. Make sure you have connected as a user with an administrative account listed in the **printer admin** parameter and that the proper directories exist as specified in the **print\$** share and the subdirectory required by the client architecture.

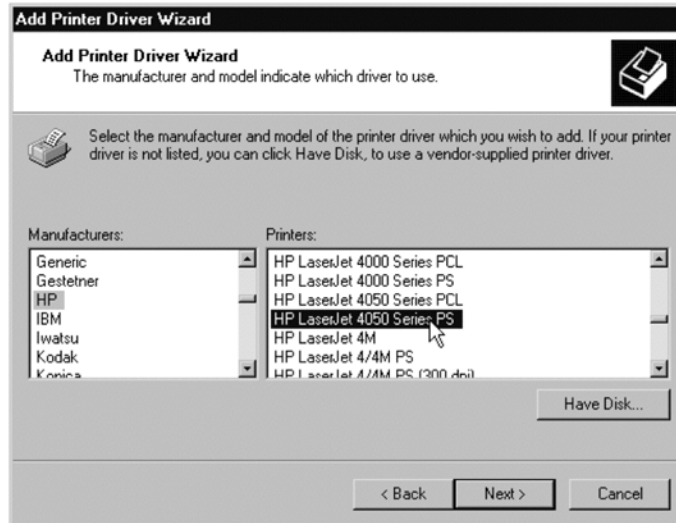
Figure 6.11 Printer Properties Dialog Box



After clicking on **New Driver** you will be allowed to select the driver (see Figure 6.12) and the files will be copied to the proper directory on the Samba

server. Samba will then update its database files to indicate which driver is associated with that printer.

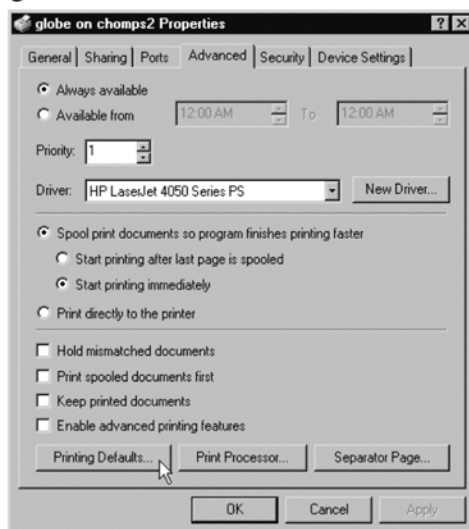
Figure 6.12 Selecting a Printer Driver



There is still additional work to do after the driver files have been installed. On a Windows server, the installation of the drivers may run a program on the server to set the initial device modes. Since these programs will not run on the Samba server, you will need to do some additional work to get these set up properly. The simplest way is to install this printer on a client machine and then open the Printing Defaults dialog box and set them from the client.

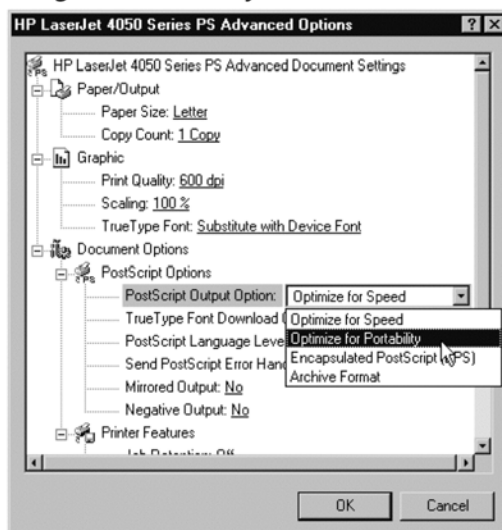
To get the default Device Modes set, open up the Printing Preferences dialog box and change the orientation to **Landscape** and click **Apply**, then change it back to **Portrait** and click **Apply** again. You can setup other default settings at this time. Click on **Advanced** and then click on **Printing Defaults**, as shown in Figure 6.13.

Figure 6.13 Selecting Printer Defaults



For a PostScript printer you may want to set the **PostScript Output Option** to **Optimize for Portability** instead of the default **Optimize for Speed**, as this tends to produce more reliable output. Figure 6.14 illustrates this setting.

Figure 6.14 Optimizing for Portability



Now print a test page from your Windows system and make sure this works. If you do not see the page printed you will need to determine where in the chain of Windows client – Samba – CUPS the error happened. Check the Samba and CUPS error logs for any obvious messages. The key to solving the problem is to check that each step in the chain works by itself.

You may want to stop the printer from printing the jobs to give you a chance to examine the file sent by Windows. If you are using CUPS click on the "Stop Printer" button on the CUPS printer settings page (Figure 6.7), or use the "disable" command. This will allow the job to be spooled but not printed. Print another test page from the Windows client and check to see if the file was spooled by clicking the "Jobs" button on top of the CUPS Printer page (Figure 6.7) or run the "lpstat -t" command.

If the job does not show up this means there was a problem either with the Windows client sending the job or Samba transferring the job to the system spooler. You need to closely examine the Samba log files (maybe increase the log level) to locate the problem. If the job does show up, examine the file in the /var/spool/cups directory (or other system spool directory) to see if it has the proper format. Try printing it with the lp command to see if that works. If that does not work, check to make sure you are using the proper printer driver.

Migrating Windows Print Services to CUPS/Samba

Now that we have our printer working from both Linux and Windows clients, we can begin the actual process of migrating our print services from the Windows server to the Linux server. Because Samba can advertise several *NetBIOS* names by using the **netbios aliases** parameter in the **smb.conf** file, you can even consolidate several Windows print servers onto one Samba server. If the Windows server is to be decommissioned, you can set up an alias on the Samba server and the clients will not need to change the name they use to reference the printer.

If the old Windows server will still remain on the network, you will have to go to each client machine and change the connection used for the printer. Since we now have point-and-print setup, you can easily delete the old printer from your Printers folder and then browse to the Samba server and select the printer from the Printers folder. Right-click on the printer and select **Connect** (or just double-click) and the drivers will be automatically installed on the client with the default settings you set.

Summary

CUPS and Samba make it simple to set up printers on a Linux server that can be used by both Linux and Windows clients. CUPS makes it simple to add printers and administer the print server. For PostScript printers, you can even use the PPD files supplied with the Windows drivers so the features that are supported on Windows are available for the CUPS server.

Samba can be set up to advertise these printers to Windows clients as well as automatically download the required drivers to the client machines. To the Windows clients, these printers look and behave just like printers that are attached to a Windows print server. Using the NetBIOS alias feature of Samba, you can even make one Linux server appear to be several Windows servers, allowing you to consolidate the number of print servers required by your organization.

Solutions Fast Track

Understanding Windows Print Services

- ☑ Windows uses the GDI as a common API for display and printing.
- ☑ Printer drivers translate the EMF output by the GDI into a format understood by the printer.
- ☑ PostScript printers use a PPD file to describe the capabilities of the printer.
- ☑ The Windows client can automatically download the proper printer driver from the server.

Understanding Linux Print Services

- ☑ UNIX systems never developed the common API for display and printing as found in Windows.
- ☑ Filters or interface scripts perform the functions of the Windows printer driver.

- ☑ Historically, the two major printing subsystems used on UNIX are BSD and SYSV. Each system uses a different set of commands to control printing.
- ☑ A newer printing system called CUPS, the Common Unix Print System, allows for management of multiple printer queue types using a web browser interface.

Sharing Samba Printers

- ☑ Set up the **smb.conf** parameters for the printing system you will be using (BSD, SYSV, or CUPS).
- ☑ Create the sections for the individual printers or use the special **Printers** section in **smb.conf** to allow your printers to be seen by Windows clients.
- ☑ Create the spool directory with proper permissions to allow the user specified by Samba to be able to create files there.

Understanding Automatic Printer Driver Downloading

- ☑ Create the entry in **smb.conf** for the **print\$** share and make sure it is writeable by printer admins.
- ☑ Create the required directory and subdirectories for the driver architectures you will support.
- ☑ Use the Add Printer Wizard from a Windows client to upload the drivers to the server and associate them with the proper printer.
- ☑ Set the default device modes and printing defaults from a Windows client.

Migrating Windows Print Services to CUPS/Samba

- ☑ Use NetBIOS aliases to have the Samba server advertise the name of a Windows server that will be decommissioned.

- ☑ If the old Windows server will remain on the network, but not be a print server any longer, you may delete old printer definitions from clients. Then use point and print to connect to the new printer location on the Samba server

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- Q:** How can I administrate my CUPS server from another system using my browser?
- A:** There are two possible issues that will prevent remote administration. Some installations of CUPS will add a **Listen** directive in the **cupsd.conf** file that will only allow connections from localhost. If you get a *Connection Refused* message from the browser, you will need to modify this line to read **Listen *:631** (or remove the line and make sure there is a line that reads **Port 631**). If you get a *Forbidden* message, this means that you have not been granted access rights. You will need to modify the appropriate section and change the **Deny From** or the **Allow From** directives. The CUPS Software Administrator’s Manual describes this process in greater detail.
- Q:** Why does my printer sometimes print jibberish followed by what looks like PostScript commands?
- A:** Most likely your Windows driver is putting some PCL commands at the beginning of the file. This causes the CUPS automatic determination of file type to fail to recognize the file as a PostScript file. Use the CUPS or the Adobe PostScript driver on the windows client.
- Q:** Why can’t my Samba spool directory be the same as the CUPS spool directory?
- A:** The setting in **cupsd.conf** for **RequestRoot** (the directory where files are spooled) and the entry in **smb.conf** for **path=** in the printers section must be

different. When your system is rebooted, cupsd will “sanitize” permissions on its spool directory. This may then make the permissions such that Samba can no longer write to this directory.

Q: Why does cupsaddsmb get errors on a printer I just created?

A: The printer may not be exported by Samba yet. You can force Samba to reread the smb.conf file by sending a HUP signal to all smbd processes. Make sure you have correctly set up Samba to share all printers (load printes = yes) and have the print\$ share properly set up.

Messaging Services

Solutions in this Chapter:

- Understanding Microsoft Messaging Services
- Understanding Linux-Based Messaging Services
- Designing Linux-Based Messaging Services
- Integrating Anti-Spam and Anti-Virus Services
- Migrating Information from Exchange to Linux
- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Messaging services provide vital communication links between company personnel and the world at large. For most businesses, e-mail is not a luxury option — it is a mission-critical priority. The loss of e-mail for a full business day will mean lost business, missed or delayed communication, business slowdown, financial losses, and frustrated employees and customers. The more time you spend learning about and planning Linux-based messaging services, the less time you will spend fixing problems and explaining outages to your boss and co-workers.

Enterprise messaging services offer multiple features to the end-users. In addition to an IMAP message store that “follows” a user to any company desktop (or laptop), a directory providing e-mail address lookups (set up in previous chapters), as well as anti-virus and anti-spam filtering, form the core sub-services for enterprise messaging. Most also provide some type of web-based e-mail client, and many incorporate groupware features such as calendars and shared folders. This chapter provides an overview of these services, lists open source solutions, and provides guidelines for design of and migration to Linux-based messaging services.

This chapter focuses on migrating from Exchange (or another messaging system with an IMAP/POP interface) to a Linux-based MTA-MDA-MAA-MUA messaging system. While most of the principles apply to non-Exchange systems, we will use Exchange in the examples, and have tested the scripts to run on Exchange servers. Although we fully expect the scripts to work with any IMAP-compatible mailstores, your mileage may vary.

The first section features an overview of a typical Exchange messaging system. These same concepts apply to many other messaging systems, but the focus of the text will be Exchange. We examine the major features and components of an Exchange 5.5 messaging system, and review the upgrades and new features found in Exchange 2000. We also discuss the differences in the directory implementation between Exchange 5.5 and Exchange 2000.

“Understanding Linux-Based Messaging Services” focuses on the components of a typical messaging system, as well as the interactions among the various components. In addition to exploring messaging system theory and key concepts, the popular mail transfer agents (MTAs) Courier-MTA, postfix, exim, and sendmail will be examined, as well as mail access agents (MAAs) including the Courier, Cyrus, and UW IMAP/POP servers. We then discuss how to design Linux-based messaging services for your organization. We explore the criteria

that form important considerations when designing a messaging system, and examine the messaging services solutions of Acme Widgets and Ballystyx Engineering.

Following the design section, you'll learn how to add anti-spam and anti-virus services. We discuss anti-spam and anti-virus solutions including dedicated gateways, shared servers, commercial appliances, and outsourcing options.

Finally, we go through the actual migration process itself. We teach you how to prepare Exchange for migration and then walk through using the scripts to transfer e-mail from an Exchange system to a Linux-based e-mail server. After you've followed all of the instructions in this chapter, you will have a Linux-based messaging system that offers the same e-mail features as Exchange without the licensing cost or other disadvantages of proprietary software.

Understanding Microsoft Messaging Services

Microsoft entered the arena of enterprise messaging and groupware with the release of Exchange 4.0 in 1996, when Exchange displaced MS-Mail as the preferred messaging system offering for large corporations. MS-Mail was difficult to administer and prone to failures. With the release of Exchange, Microsoft offered corporations a full-featured multi-master LDAP (Lightweight Directory Access Protocol)-capable directory service paired with a JET-derived enterprise mailstore database service. MS-Mail servers typically supported 100 or more users, but Exchange servers could support thousands of users with appropriate hardware.

Exchange is a direct competitor of Novell Groupwise, Lotus Notes, Oracle Messaging, and other proprietary messaging products. Today, Exchange and Notes dominate the high-end messaging and groupware marketplace among Fortune 100 companies, with a plethora of open source and proprietary solutions servicing the small-to-medium business market niche.

Exchange offers user mailboxes, e-mail groups, shared public folders, IMAP and POP interfaces, shared contacts, user and group calendars, flexible database storage, and almost any feature required of an enterprise messaging and groupware system. While Exchange has the advantage of being a mature, full-featured messaging system, a drawback of Exchange is the costly integration for anti-virus and anti-spam. While no single open source solution can match the features of Exchange 2000 / Active Directory in an integrated package, most of the

Exchange features and capabilities are available using a combination of open source messaging software.

Understanding Exchange 5.5 Messaging

A Microsoft Exchange 5.5 server is actually a handful of integrated services. Table 7.1 lists the Exchange 5.5 component services with a brief description of each one.

Table 7.1 Exchange 5.5 Component Services

Service	Function
System Attendant	Monitors all the other services
Directory Service	Provides the Exchange directory services
Information Store	Manages the database that stores the contents of public folders and user mailboxes
Message Transfer Agent	Transfers messages between Exchange servers
Internet Mail Service	Transfers messages between hosts using SMTP (Simple Mail Transfer Protocol)

Exchange 5.5 servers are grouped into sites. Servers within a site communicate via an always-available high-bandwidth connection, such as servers that are located on a LAN (Local Area Network) and possibly high-bandwidth WAN (Wide Area Network) subnets. In Windows 2000 parlance, these sites are collections of IP (Internet Protocol) subnets. Within an Exchange organization, each site must have a unique name, and share the same organization name.

Exchange 5.5 connectors link Exchange sites with other sites, or with a foreign transport system, such as SMTP, Notes, MS-Mail, Fax, or X.400 systems. These connectors allow Exchange to separate services into specific additional address types, and offload processing of these types to the connector software.

One of the most popular connectors, the Internet Mail Service (included with both the Standard and the Enterprise versions of Exchange Server), offers a significant number of options for configuring SMTP communications, including to/from the Internet. Figure 7.1 shows Acme Widget's Internet Mail Service (IMS) configuration.

Figure 7.1 Acme Widgets IMS Configuration

When two Exchange 5.5 sites are linked together using a site or other messaging connector, they may exchange directory replication information. Within a site, directory information is replicated via RPC (remote procedure call). Between sites, directory information is replicated via e-mail messages communicated by directory replication connectors.

Exchange supports a number of protocols for access to e-mail (and groupware) information. Table 7.2 lists these protocols and describes their usage.

Table 7.2 Exchange 5.5 Protocol Support

Protocol	Description
MAPI	Messaging API (Application Program Interface) is used by Outlook and provides mailbox and groupware features
IMAP	IMAP is a full-featured cross-platform mailbox access protocol
POP	POP is a limited-featured cross-platform mailbox access protocol
HTTP	HTTP (Hypertext Transfer Protocol) allows web browsers to access messages and calendar/groupware via OWA (Outlook Web Access)
LDAP	LDAP enables cross-platform access to Exchange directory information

Exchange 5.5 supports *active-passive* clustering, or what is called a *failover* clustering. To take advantage of clustered Exchange 5.5 servers, you must use a cluster-capable form of storage such as a SAN (Storage Area Network) or a shared SCSI (Small Computer Systems Interface) bus. One of the Exchange servers must be designated a *passive* node that simply waits for the other server to fail. *Active-active* or *load balancing* clustering (both servers are actively working) is supported in Exchange 2000, but is not supported in Exchange 5.5

Understanding Exchange 2000 Messaging

Exchange 2000 offers the same core features as Exchange 5.5, but most features are improved or upgraded. Exchange no longer uses its own directory service, but instead requires Active Directory. All information that was stored in the Exchange 5.5 directory is stored in Active Directory for Exchange 2000. When Exchange 2000 is installed, it simply updates the Active Directory schema to e-mail- and groupware-enabled directory objects, and provides interfaces to configure mail storage and message transfer.

The mailstore in Exchange 2000 uses a more scalable architecture, with each server handling storage group(s) instead of a single public folder / private mailbox store on each server, as was the case with Exchange 5.5. This allows for greater flexibility in the areas of storage, backup, restore, and clustering. Mailbox storage may be easily split up across multiple smaller-sized databases, which can be stored on different storage devices, use different backup schedules, and be restored more quickly than a single large database.

The OWA component is significantly improved, and more full-featured than Outlook Web Access in Exchange 5.5, particularly when Internet Explorer is used as the web browser. The additional features in OWA 2000 include support for drag and drop, WYSIWYG message composition, and enhanced display of embedded objects.

In Exchange 5.5, the internal addressing scheme is primarily X.400 convention, with all user mailboxes possessing an X.400 address. In Exchange 2000, the primary addressing mechanism is SMTP, so SMTP is used as the primary communication mechanism.

Active Directory handles the replication, trust, site connections, and other features formerly managed by Exchange 5.5 directory services. In Exchange 2000, sites are now part of active directory, with replication being managed by Windows 2000 domain controllers.

Understanding Linux-Based Messaging Services

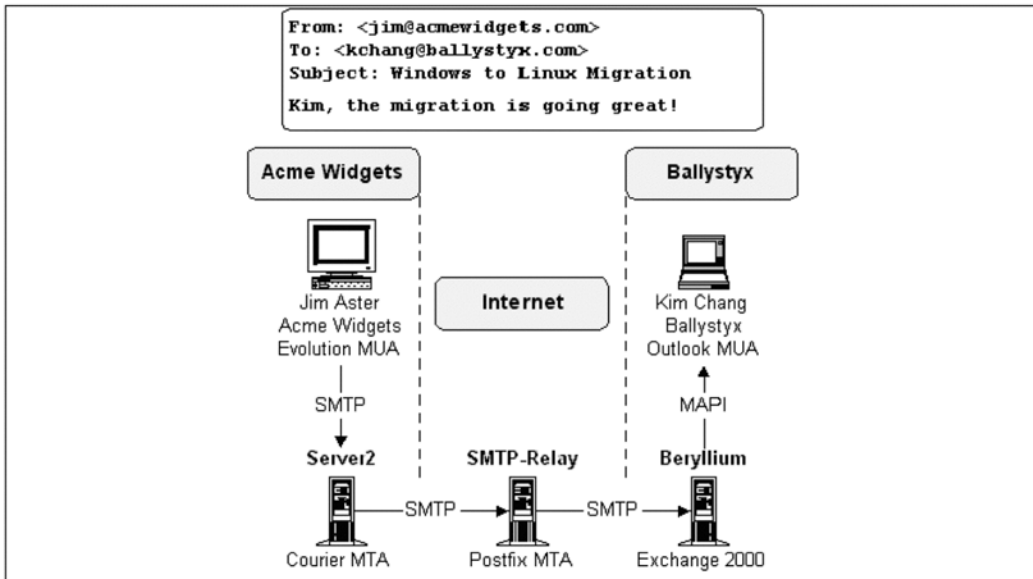
Most open source messaging systems are comprised of an integrated, yet modular, system comprised of multiple component applications. These messaging components are responsible for transmitting, receiving, delivering, and providing access to e-mails. In smaller companies, these components are typically part of a single messaging suite, such as the Courier-MTA suite. In larger companies, multiple components are often “glued” together to form a customized messaging solution tailored toward the specific needs of that organization.

In this section we will examine the lifecycle of an Internet e-mail and then examine the individual components of a messaging system that allow this process to happen. We also list the open source server applications that form the basis of Linux-based messaging services, and present an overview of each application.

Sending and Receiving Internet E-mail

To better understand messaging systems, we will examine sending and receiving Internet e-mail. In other words, we will follow the lifecycle of an Internet e-mail message from the point that a user clicks **Send** to the point that the e-mail arrives in the recipient's Inbox.

The process of sending an Internet e-mail message begins with an MUA (mail user agent) submitting an e-mail to an MTA. In the example in Figure 7.2, Jim Aster at Acme Widgets has composed an e-mail using Evolution, and is sending the e-mail to Kim Chang at Ballystyx Engineering. In this figure, Acme Widgets is in a post-migration state, and is using a Linux-based MTA and MUA. Ballystyx Engineering is in a pre-migration state, and is still using an Exchange / Outlook messaging system.

Figure 7.2 Sending and Receiving Internet E-mail

The actions that take place in this diagram can be broken down to seven basic steps:

1. Jim Aster composes an e-mail and clicks **Send**.
2. Evolution submits the e-mail to SERVER2.
3. Courier receives the e-mail via SMTP.
4. Courier determines that the e-mail is for a non-local SMTP recipient, and forwards the e-mail to SMTP-Relay.acmeISP.net.
5. Acme ISP's SMTP relay machine receives the e-mail and queues it for outbound delivery
 - Jim's e-mail in the outbound delivery queue is picked up by the MTA for delivery.
 - The MTA performs a DNS (Domain Name System) lookup for the MX (Mail Exchanger) record of ballystyx.com and finds that it maps to beryllium.ballystyx.com.
 - The MTA connects to beryllium.ballystyx.com and transfers Jim's e-mail via SMTP to Exchange 2000.
6. Exchange delivers the e-mail to the mailbox of Kim Chang.

7. A few minutes later, Kim Chang opens Jim's e-mail using Outlook's MAPI connection to beryllium.

These actions take place tens of billions of times every day, all across the planet. The number of MTAs involved and the software that serves each function may vary, but Figures 7.2 (above) and 7.3 (below) summarize the components, communication, and specific steps that take place in order for this process to succeed from start to finish.

Now that we understand how to send and receive Internet e-mail, let's take a closer look at the components of a Linux messaging server.

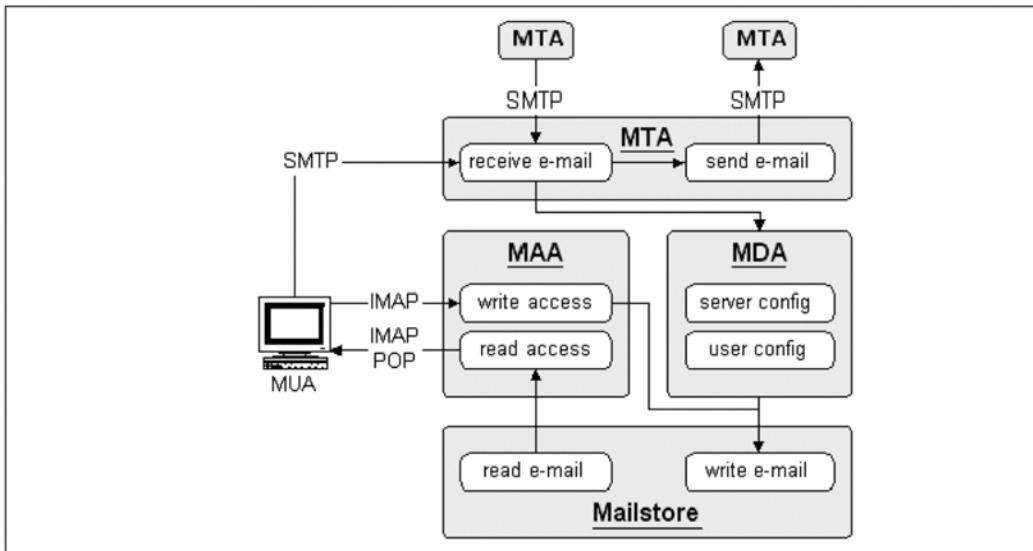
Understanding Linux Messaging System Components

A messaging server is not just one component, but many. Receiving mail, sending mail, delivering mail, and interacting with mail clients involves a number of processes that must seamlessly work together for the messaging system to function properly as a whole. These components and short descriptions are listed in Table 7.3. We will be referring to these terms frequently throughout the chapter.

Table 7.3 Messaging System Components

Component	Abbreviation	Short Description
Mail user agent	MUA	E-mail client for reading, sending, and deleting messages
Mail transfer agent	MTA	Host-to-host message transfer service
Mail delivery agent	MDA	Delivers messages to local mailstore
Mail access agent	MAA	Responds to MUA requests to view messages
Mailstore		Storage mechanism for messages and mailboxes

Figure 7.3 shows a diagram of a modern Linux-based e-mail server and the MTA, MAA, MDA, and mailstore components that comprise it. The MUA actions, and other (possible) MTAs are also shown.

Figure 7.3 E-mail Server Components

We will examine each of these components and protocols more closely in the sections below.

Mail User Agent (MUA)

A mail user agent is the client portion of the messaging system that interfaces with an end user. In many cases this is referred to simply as a *mail client* or *e-mail client*. These terms are mostly interchangeable and are commonly used to refer to applications such as Evolution, Mozilla, Thunderbird, KMail, Outlook, or Pine. Another type of mail client is a web-based client, typically installed as a cgi-bin add-on to Apache. Two popular open source webmail applications are SquirrelMail and SqWebMail, described later in this chapter. Simply stated, an MUA displays the contents of messages in the user's mailbox.

The MUA communicates with the MTA to send messages, and communicates with the MAA to access the mailbox. In Linux (and nearly all) TCP/IP (Transmission Control Protocol/Internet Protocol) messaging systems, SMTP is used for e-mail submission and IMAP (or POP) is used for mailbox access.

Most modern MUAs support the use of encryption to ensure that the contents of e-mails are not decoded during transit across the network. In many cases, e-mail will contain confidential or sensitive information, so it makes sense to use encryption across network transit points. The StartTLS (Start Transport Layer

Security) mechanism has become the standard way to implement connection encryption when communicating with an e-mail server.

Most e-mail clients support an offline configuration that allows for limited operation while disconnected from the network. In many companies, the e-mail is stored on the server and may be mirrored on a laptop computer for viewing during disconnected operation. In other companies, the e-mail is downloaded to the hard drive on the laptop. This can be problematic if the laptop is lost, stolen, or damaged. The e-mail is usually lost forever because it hasn't been backed up, and the loss of this (and other) information will likely have a serious impact on the productivity of the ex-owner of the laptop. In addition, when mail is downloaded to a local client, it can only be viewed on that particular client, and web-mail MUAs will show an empty mailbox.

Most MUAs support address lookups via a directory server (such as OpenLDAP). This makes finding people, e-mail addresses, and contact information much easier, particularly for employees of large multi-site companies. Because the migration scripts included with this book populated your OpenLDAP directory with e-mail addresses in Chapter 4, this address lookup feature is already available to your users when they switch to Linux-based messaging services.

There is a significant variety of open source MUAs available for Linux, and Table 7.4 lists popular MUAs and their websites.

Table 7.4 Popular Linux MUAs and Websites

MUA	Website
Evolution	www.novell.com/products/evolution/
Mozilla	www.mozilla.org/products/mozilla1.x/
Thunderbird	www.mozilla.org/products/thunderbird/
KMail	kmail.kde.org/
Balsa	www.newton.cx/balsa/
Pine	www.cac.washington.edu/pine/

More detailed information about Linux MUAs can be found in Chapter 11, "Inside the Linux Desktop." A comprehensive list of Linux MUAs is located at www.linuxmafia.com/faq/Mail/muas.html.

Mail Access Agent (MAA)

An MUA must have a way to access the contents of the mailstore. This is accomplished via the MAA. The access method might be via a file share, or via a protocol such as IMAP or POP. The MUA instructs the MAA to retrieve, delete, insert, and/or move e-mails, and the MAA manages read and write operations with the mailstore.

Courier-IMAP, UW-IMAP, and Cyrus-IMAP are three popular open source MAAs. In general, Courier-IMAP is the recommended IMAP MAA, whether or not the other applications in the Courier suite are being used. UW-IMAP has a poor security record and doesn't work with maildirs, and Cyrus-IMAP uses an unusual mailstore methodology. All of these MAAs service clients using IMAP and/or POP. We discuss the merits and characteristics of IMAP and POP in the sections below.

Understanding Post Office Protocol

Post Office Protocol is a predecessor to IMAP and is still a widely used network protocol for mailbox access. In this book, when we refer to POP, we are referring to POP version 3, defined in RFC 1939. In most environments, POP is used to download e-mail from a server to local storage, and deletes the e-mail from the server following successful download.

POP does not offer support for server-based folders or for anything more complex than simply retrieving and deleting e-mail messages. In many cases, these capabilities are enough for simplistic messaging services.

Understanding Internet Mail Access Protocol

IMAP allows for access to a mailbox on a remote server. When we refer to IMAP in this book, we are referring to IMAP version 4 described in RFC 3501. While POP only allows for e-mail download and deletion, IMAP enables complete management of remote mailboxes, and is considered the premier protocol for enterprise-class access to user mailboxes. It is optimal for server-side storage of e-mail, hierarchical subfolder organization, and access from multiple computers, multiple locations, and full-featured disconnected operation.

While IMAP features more capabilities than POP, IMAP is more complex to deploy. It is important to keep in mind that POP and IMAP serve slightly different types of requirements. In many cases, it is easy to quickly deploy a POP server and configure clients to use POP to download the e-mail to a local hard drive. In fact, some older applications do not support IMAP.

However, for full-featured access to mailboxes, support for shared folders, and support for disconnected operation, IMAP is worth the extra effort, and will add value to messaging services. Storing mailstore data on the mail server enables webmail and simplifies backup and restoration of messages. For information about installing and configuring Courier-IMAP, navigate to www.courier-mta.org/imap/INSTALL.html.

Mail Transfer Agent (MTA)

A mail transfer agent is primarily responsible for transferring messages from one host to another – in other words, sending and receiving e-mail across a network. On the Internet and almost any other TCP/IP network, the protocol to accomplish this is SMTP. SMTP and Internet e-mail formatting information is described in RFCs 821, 2821, 2822, and 2487.

To send an e-mail using SMTP, the host initiates an e-mail transfer by contacting another host on TCP port 25. If an SMTP server is running, the client and server engage in a dialog. A typical MUA - MTA conversation at Acme Widgets between Evolution, running on WIDGET1, and Courier-MTA, running on SERVER2, is illustrated in Figure 7.4. The text that the client (WIDGET1) is sending is in **bold**.

Figure 7.4 SMTP Conversation

```

SERVER2: 220 server2.acmewidgets.com ESMTP
WIDGET1: EHLO widget1
SERVER2: 250-server2.acmewidgets.com Ok.
SERVER2: 250-STARTTLS
SERVER2: 250-PIPELINING
SERVER2: 250-8BITMIME
SERVER2: 250-SIZE
SERVER2: 250 DSN
WIDGET1: STARTTLS
SERVER2: 220 Ok
    ( The rest of the conversation is encrypted )
WIDGET1: MAIL FROM: <sender@acmewidgets.com>
SERVER2: 250 Ok
WIDGET1: RCPT TO: <recipient@ballystyx.com>
SERVER2: 250 Ok
WIDGET1: DATA

```



```

SERVER2: 354 Ok
WIDGET1: ( e-mail message contents )
SERVER2: 250 Ok
WIDGET1: QUIT
SERVER2: 250 Ok

```

As you can see in the conversation, the StartTLS command initiates encryption. Evolution knew that encryption was supported because SERVER2 advertised the capability with the response

```
250-STARTTLS
```

Most modern e-mail clients and servers support encryption. However, getting encryption to work may be as easy as clicking a checkbox or as much work as dealing with a Certificate Authority (CA). For MUAs, encryption is usually enabled by simply checking a **Use SSL/TLS** checkbox. For MTAs, encryption usually requires the installation of a server certificate and the reconfiguration of various servers to actually use the certificate and enable encryption. Managing OpenSSL certificates is covered in Chapter 9, “Web Services.”

An e-mail message has only one sender, but may have multiple recipients. If an e-mail is addressed to multiple recipients at the same domain, most MTAs will deliver only one e-mail message to the domain. For example, if jim@somedomain.com sent an e-mail that was addressed to john@example.com and mary@example.com, the MTA at somedomain.com would only need to send one copy of the e-mail to the mail exchanger for example.com, and the receiving MTA at example.com would manage delivering a copy of the e-mail to the mailboxes of john and mary.

Attachments in an e-mail message are managed through MIME, or Multipurpose Internet Mail Extensions. MIME separates an e-mail into multiple sections. Each section may have a different type of content format. More information about MIME is available at <http://en.wikipedia.org/wiki/MIME> and in RFCs 2045, 2046, 2047, 2048 and 2049.

MTAs do their work by using multiple queues for temporary message storage. All modern MTAs utilize at least an inbound, an outbound, and a work queue. Most MTAs also use a combination of deferral, quarantine, trash, and/or logging queues to categorize e-mail for processing. When an MTA receives an e-mail addressed to a user on the local server, the MTA transfers the message to the Mail Delivery Agent to be copied to the mailstore.

Mail Delivery Agent (MDA)

The mail delivery agent is responsible for delivering messages to a mailstore. An MDA is sometimes referred to as a local delivery agent (LDA). In this book we will refer to the agent that delivers the message to the mailstore as the MDA. Most MTAs include their own MDA, but can easily be configured to use another MDA if desired.

Many MDAs feature a rules-based filtering mechanism that can be employed site-wide and/or on a per-mailbox basis. Site-wide configuration settings are usually stored in `/etc/`, while user-specific settings (such as moving certain incoming mail to a specific mail folder) are usually stored in the user home directory or mail spool directory.

Procmail and maildrop are two popular mail delivery agents. Both allow for e-mail filtering, although the language for procmail *recipes* is considered cryptic. Maildrop, the MDA used by Courier-MTA, features an easier syntax that is similar to scripting languages.

Fetchmail is an MDA with MTA-like functions. Fetchmail retrieves e-mail from POP or IMAP mailboxes and forwards the e-mail to an SMTP MTA. Fetchmail features a graphical configuration tool, `fetchmailconf`, which assists with the many configuration options of fetchmail. Fetchmail is even capable of managing e-mail for an entire domain through a single mailbox, although that is not necessarily the best use of fetchmail. More information about fetchmail is available at www.catb.org/~esr/fetchmail/ or by typing **man fetchmail**.

Mailstore

In addition to sending, receiving, and delivering e-mail, there is the important consideration of how to store the e-mail data on a Linux mail server. In most Linux servers, this is accomplished using file-based storage, although some products may use a database to store e-mail. While Exchange uses a JET-derived database, open source Linux applications use two different formats for e-mail message storage: `maildir` and `mbx`.

Mbox/Mbx Mail Storage Format

The mbox methodology is the historical way to store mail on UNIX-based mail servers. Individual messages are simply concatenated together into one file, with a marker placed between messages. In mbx, this string contains metadata to enhance indexing performance. This simplistic methodology is appropriate for a

moderate amount of individual e-mail on a single-user workstation, and may be used as the local storage mechanism by a number of e-mail clients.

While mbox and mbx are fine for client e-mail, they are not the preferred methodology for robust e-mail storage when multiple processes need concurrent read/write access to the mailstore. Because only one process can write to the mbox file, this storage methodology scales poorly due to locking problems. This potential problem is particularly apparent when mailbox data is housed on network file shares (which may or may not properly enforce locking rules) and multiple processes are trying to write to the single file. In addition, a power failure in the mailstore server during disk write operations will often corrupt (at least) the last message in an mbox file, sometimes the entire mbox file.

Maildir Mail Storage Format

Maildir is a more recent mail storage format than mbox. Maildir was invented by the author of Qmail, and the first maildir implementation was introduced in Qmail. One of the most significant differences between mbox and maildir mailstores is that a maildir mailstore stores e-mail messages as separate files. No locking is required, which means that this delivery format can work even when there is faulty lock enforcement in the underlying network file protocols or applications. Multiple processes can write to maildir mailstores at the same time without locking issues or lost/corrupted e-mail.

More information about the file delivery and manipulation mechanisms used in maildir can be found in Professor Bernstein's original documentation about maildirs at www.qmail.org/man/man5/maildir.html. A more comprehensive discussion about maildirs that includes maildir quotas may be found at www.courier-mta.org/maildir.html.

Database Mail Storage Format

Database-style mail storage is primarily used in commercial proprietary products, such as Microsoft Exchange or Oracle Messaging. Most open source enterprise messaging applications use maildir or mbox formats, although Cyrus-IMAP uses its own database-like mailstore. While database mail storage offers a number of potential advantages, it also increases the complexity of storage and introduces potential hassles not present with the relatively simplistic file-and-directory-based mail storage mechanisms. For now, maildir seems to be the preferred open source mail storage format for enterprise open source messaging.

Exploring Open Source Messaging Server Software

The history and variety of open source messaging server software begins with Sendmail in 1980, and follows in the 1990's with Postfix, Exim, Qmail, and Courier. Now, numerous options for sending, receiving, delivering, and accessing e-mail using open source software are available to choose from. In this section we will explore some of the most popular messaging software.

Historically, there have been numerous ways to transfer e-mail between hosts. Sendmail was the first generic MTA. It was followed by several others that never gained popularity. Qmail followed as a secure, more-efficient Sendmail replacement. Qmail came up with a more efficient mail storage format that addressed several issues with the existing way that e-mail was usually saved. However, Qmail was a very plain, bare-bones MTA. Postfix was created to solve the cruft, cryptic configuration syntax, and other issues with Sendmail, but also to provide a robust MTA that implements most modern features expected of the MTA. Finally, several e-mail-related projects combined and became the Courier MTA.

Sendmail

A discussion of open source messaging server software needs to begin with Sendmail. Sendmail is the granddaddy of all MTAs. The initial version of Sendmail was written in 1980, long before SMTP became the global e-mail transfer protocol. At the time, e-mail messages moved around on a variety of different networks and message formats. Sendmail's claim to fame was its ability to transfer messages between different networks by reformatting the message to fit the format of each network.

Although Sendmail is still widely used, this is not because of simplicity of operation or ease-of-use. Sendmail uses a complex, intimidating, configuration file. Its complexity is legendary, and is not easily tackled by novices. Sendmail's configuration file is really a mini-programming language of its own; it is a low-level script that specifies how Sendmail processes e-mail addresses and headers, step by step. A post to a Usenet newsgroup in 1996 showed how Sendmail's configuration file can be used to make Sendmail solve the Towers of Hanoi puzzle.

Fortunately, modern versions of Sendmail include a macro preprocessor. Instead of coding Sendmail scripts by hand, Sendmail's configuration is described using a set of high-level configuration directives. A macro preprocessor then translates the directive into Sendmail's native configuration file.

Furthermore, Sendmail is already included as a standard part of all major Linux distributions. It is usually supplied with a generic configuration, so in most cases almost no configuration is required at all. However, for small businesses migrating to a Linux infrastructure, we do not generally recommend the choice of Sendmail. There are better (simpler) choices out there, and most Linux distributions offer another MTA, in addition to Sendmail, as the default.

For more information on Sendmail, visit www.sendmail.org. For information about Sendmail support services and graphical configuration tools, visit www.sendmail.com.

Qmail

Qmail is an interesting MTA. Its popularity is slightly hampered by its unusual licensing terms, but it still enjoys a favorable security record. At one time the author of Qmail offered \$500 to anyone who could find a security flaw – the bounty went unclaimed.

The Qmail MTA was written by Professor Daniel Bernstein in the early 1990's. Qmail is a direct opposite of Sendmail in many ways. Sendmail runs as a single monolithic process that implements all functions that Sendmail performs. Qmail is a collection of multiple small modules that work together, with each module dedicated to a single task. Sendmail uses a single configuration file with cryptic syntax, while Qmail stores its configuration data as separate, plain-text files.

Qmail's major contribution to MTA technology is the development of the maildir format for e-mail storage. Traditionally, all messages in a folder were saved as a single file. This mail storage format, called mbox, was sufficient for the early days of the Internet, but mbox's limitations began to show as e-mail grew in popularity. Only one process could update the folder at any time, so locking was necessary to coordinate folder access between competing processes, and traditional file locking did not work reliably with network-based filesystems. To update a folder, a process has to essentially rewrite either the entire file, or most of it, from scratch. If the process gets interrupted before it finishes, the result is a corrupted mail folder.

Qmail's maildirs place each message into a separate file. A specific naming convention determines the filename of each message, and there is a well-defined mechanism for multiple processes to update the contents of the mailbox at the same time without stepping on each other's toes and without needing any form of locking.

Maildirs proved to be very popular, and are now supported by Courier, Postfix, and Exim MTAs. With a cooperating MDA, such as maildrop or procmail, Sendmail can also deliver to maildirs.

Prof. Bernstein also came up with a new quick mail transfer protocol, dubbed *QMTP*, as an alternative to SMTP. Although maildirs are now a fixed staple in most MTAs, QMTP has not caught on, and today is still found only in Qmail.

Qmail is a very small MTA. It implements only the bare minimum of SMTP. Over the last twenty years SMTP has significantly evolved. It acquired many extensions to its original functionality, but Qmail hasn't kept up. Prof. Bernstein's last version of Qmail, 1.03, was released in 1993. Qmail lacks the advanced SMTP functionality expected from modern MTAs, such as encryption and authenticated SMTP.

One major drawback to using Qmail in today's environment is that Qmail's design requires Qmail to accept all mail addressed to the local e-mail domain. If the destination mailbox does not exist, the Qmail server attempts to return the e-mail message to its purported sender as *undeliverable*. The major problem here is spam. Junk messages usually carry undeliverable return addresses. Qmail will be unable to return the message to the sender, and the message will remain in Qmail's mail queue for some time before it times out, and gets deleted. Large amounts of undeliverable junk messages could have a significant impact on Qmail's performance.

Although Qmail hasn't been updated in some time, it has a large user base. A number of third-party patches and add-ons to Qmail address some of its shortcomings. Qmail is still often considered in security-sensitive environments. Because it's so small, there's less of a chance of an exploitable defect in Qmail. In general, the more complex the feature set of a software application, the more likely the software has bugs. Qmail's small size also makes it a popular choice for some firewall environments.

Navigate to www.qmail.org for more information on Qmail.

Postfix

Postfix was initially developed by IBM in 1997. It was originally called "VMailer", then "IBM Secure Mailer". Postfix was spun off early in its development cycle, and now exists independently of IBM. Postfix was also originally intended to be an alternative to Sendmail, but is actually more similar to Sendmail than to Qmail. Postfix continues to be actively developed today, and is the default MTA in many modern Linux distributions.

Postfix was designed to offer the robustness of Sendmail, with improved speed, simpler configuration, and designed from the start with security in mind. The result has been what many believe to be the most commonly-installed Linux MTA.

Postfix includes several spam filtering options, and a variety of options for implementing virtual mailboxes. In the beginning, mailboxes and UNIX system accounts were one at the same – if you had a UNIX account, you had a mailbox and an e-mail address with the format *username@domain*. When e-mail became a separate service, there was no longer a need to have a system account for each mailbox. An e-mail server became a separate system with mailboxes became accessible via IMAP or POP3 only. With Qmail or Postfix, virtual mailboxes can be created without an actual system account by adding the account information into a file. The list of accounts in the file then gets compiled into a small database file. Postfix also offers the option of using an LDAP server, a MySQL server, or a PostgreSQL server instead of files.

Visit www.postfix.org for more information on Postfix.

Exim

The Exim MTA was developed by the University of Cambridge. Exim specializes in its mail filtering and mail checking abilities, using its own special-purpose mail filtering language. Like Postfix, Exim offers support for LDAP, MySQL, and PostgreSQL databases.

One interesting filtering feature in Exim (it is also available in Postfix) is its *callout* ability. Before accepting a message, Exim can examine the sender's e-mail address, and attempt to contact the sender's mail servers to verify that the address exists. This slows down overall mail delivery somewhat, but mostly eliminates spam with forged, nonexistent return addresses. Like Postfix, Exim is easily integrated with SpamAssassin.

Another interesting feature of Exim is that it can use embedded Perl scripts for some of its tasks. Exim can probably be described as an MTA for programmers, in the sense that programming skills allow an e-mail administrator to take full advantage of Exim's capabilities.

See www.exim.org for more information on Exim.

Courier Suite

Courier is an entire modular messaging suite that includes an MTA (Courier-MTA), webmail MUA (SqWebMail), MDA (maildrop), MAA (Courier-IMAP),

web-based Administration (courierwebadmin), LDAP integration, and calendaring / groupware features. Each application of the suite is a stand-alone product, but they share a common code base and work together in an integrated fashion. Courier is configured to work with maildir mail storage.

Table 7.5 lists the components of the Courier suite.

Table 7.5 Courier Suite Components

Component	Description
Courier-MTA	MTA for host-to-host message transfer
maildrop	local MDA, similar to procmail
Courier-IMAP	MAA for IMAP/POP mailbox services
SqWebMail	MUA web-based e-mail and calendar access (requires Apache cgi-bin)
Webadmin	Web-based administration module for Courier (requires Apache cgi-bin)
Courier-MLM	Mail list manager
Courier-analog	Usage reports

Courier features an integrated solution for e-mail transfer, storage, client access, mailing list management, and web-based access services. It contains a web-accessible calendar solution. Courier also features a web-based management console for those who prefer GUI (graphical user interface) administration. On top of all of this, the entire Courier suite is easily secured via SSL/TLS encryption. Courier meets Acme Widgets' current needs, and will meet all medium-term foreseeable requirements. Acme Widgets may consider the Courier fax gateway option for possible deployment at a later time.

The feature set and ease of installation makes the Courier suite a good choice for Acme Widgets. Since Acme Widgets has limited messaging requirements and limited budget for IT work, they get all the functionality they need without the complexities (and extra work) of trying to get multiple MTA, MAA, MDA, and webmail packages to work with each other.

Courier-MTA

The central part of Courier consists of a typical MTA. Courier-MTA receives new e-mail messages, places them in an internal queue, schedules them for delivery, and reschedules messages that could not be immediately delivered.

Courier provides several controls for selectively accepting and rejecting messages. There are several pre-defined filters that check messages for correctness. It is also possible to install a Perl script or a separate program that examines messages and selectively rejects unwanted e-mail.

Courier, like Exim and Postfix, can use LDAP, MySQL, or PostgreSQL. See www.courier-mta.org for a general introduction to Courier.

Maildrop

The maildrop MDA comes with Courier, but Courier-MTA can also use any other MDA. Maildrop is similar to procmail, but uses a simpler, Perl-like e-mail filtering instructions.

Maildrop can be installed either as the global MDA used by Courier-MTA to deliver to every account, or selectively on a per-account basis. Courier-MTA can also used maildrop directly as a mail filtering control; however this is an advanced, complicated task that requires some amount of programming. For most situations maildrop's role as an ordinary MDA will be sufficient.

See www.courier-mta.org/maildrop/ for more information on maildrop.

Courier-IMAP

The Courier-IMAP server provides IMAP and POP3 access to local mailboxes. Like SMTP, IMAP has evolved over the years, and has acquired many extensions to the basic functionality on top of the IMAP core. Courier-IMAP implements most of the popular IMAP extensions, such as access control lists (ACLs). It is possible to set up shared mail folders in Courier-IMAP, accessible by multiple accounts in accordance to rules configured by IMAP ACLs. This can be used to implement basic groupware functions, by using shared folders to exchange memos and moderately sized files.

Like the other Courier applications, Courier-IMAP may be installed independent of the Courier suite, and may be used with any maildir mailstore. A popular combination in larger businesses is using Postfix to send/receive e-mail and Courier-IMAP to provide e-mail access services for clients.

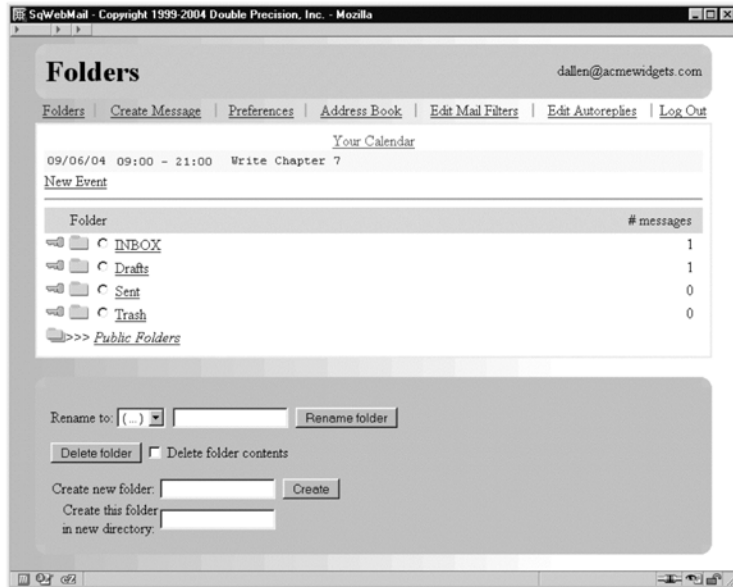
See www.courier-mta.org/imap/ for more information about Courier-IMAP.

SqWebMail

Courier's SqWebMail is full-featured webmail server, with calendar / groupware functionality. The interface supports user mailbox folders and shared folders. Address books, spell-check, calendaring, mail filtering, and even GnuPG?

cryptography are supported! These display well on any browser, and do not require Javascript support. Figure 7.5 shows a screenshot of SqWebMail.

Figure 7.5 Courier SqWebMail Webmail Client



SqWebMail is very fast. Other webmail servers access messages by establishing a background connection to an IMAP server and downloading e-mail messages using IMAP before formatting them for display. SqWebMail accesses maildirs directly, without using IMAP.

SqWebMail also offers an option to generate basic maildrop filters using a simple web interface. This enables creation of basic e-mail filters without knowledge of programming.

SqWebMail, and webadmin, require a cgi-bin capable server, such as Apache. See www.courier-mta.org/sqwebmail/ for more information on SqWebMail. The website includes many screenshots.

Webadmin

Courier's webadmin is a web-based interface that provides access to most configuration settings. Following updates to the server's configuration, webadmin automatically restarts affected Courier module(s). Figure 7.6 shows a screenshot of Courier's webadmin.

Figure 7.6 Courier Webadmin Management GUI

Courier-MLM

Courier-MLM is a basic mailing list manager. To set up a mailing list, create a separate account on the Courier server under the mailing list's name, then install Courier-MLM as the account's MDA, following the directions in Courier-MLM's documentation. The result is a basic mailing list.

Courier-MLM will be adequate for most internal company mailing lists. Situations that require an advanced mailing list processor will need to use an external mailing list manager with Courier.

Courier-analog

Courier-analog is Courier's log analyzer. This module is not included in the main Courier distribution. As part of their normal operation, Courier's many servers send messages to the syslog daemon, which saves them to a file (usually `/var/log/messages` or `/var/log/maillog`). Courier-analog is a Perl script that reads the log file and generates usage reports for Courier-MTA and Courier-IMAP.

Courier-analog summarizes SMTP, IMAP, and POP3 usage in several different ways. IMAP and POP3 usage is summarized by the connecting network, time and amount of downloaded e-mail messages. SMTP traffic is summarized by the sender or receiver, as well as overall usage, time, and error messages. Courier-analog generates plain text or HTML (Hypertext Markup Language) reports.

Designing Linux-Based Messaging Services

Designing Linux-based messaging services primarily consists of determining the type, size, and placement of MTA(s), MAA(s), and mailstores, as well as determining the software applications to use and the configuration of each application. The design must also include a path for sending and receiving Internet mail. These choices will be determined by the messaging requirements identified in the requirements document, and may be constrained by budget, time, or other factors.

For a small company like Acme Widgets, this choice is easy. Because there is only one server and one site, Sam simply installs and configures the Courier suite on the Linux server, and the messaging system deployment phase has been completed in a few minutes. In a larger company such as Ballystyx Engineering, the situation is not so simple and unambiguous. There are multiple servers, multiple sites, and many users. Because e-mail performance is mission-critical and there is sufficient budget to address design considerations, it makes sense to carefully review messaging system requirements to ensure that the design will meet the current (and future) needs.

From a scalability perspective, the primary goal of any company designing Linux-based messaging services should be to design a messaging system that can process the volume of e-mail the company is presently experiencing, and will manage the expected volume of e-mail and storage for the next 2–3 years (including the inevitable spam and viruses). From a functionality and reliability perspective, the messaging system must provide the appropriate messaging features to the users, provide sufficient performance, and be as reliable as is feasible within the budget provided.

Determining the Internet E-mail Architecture

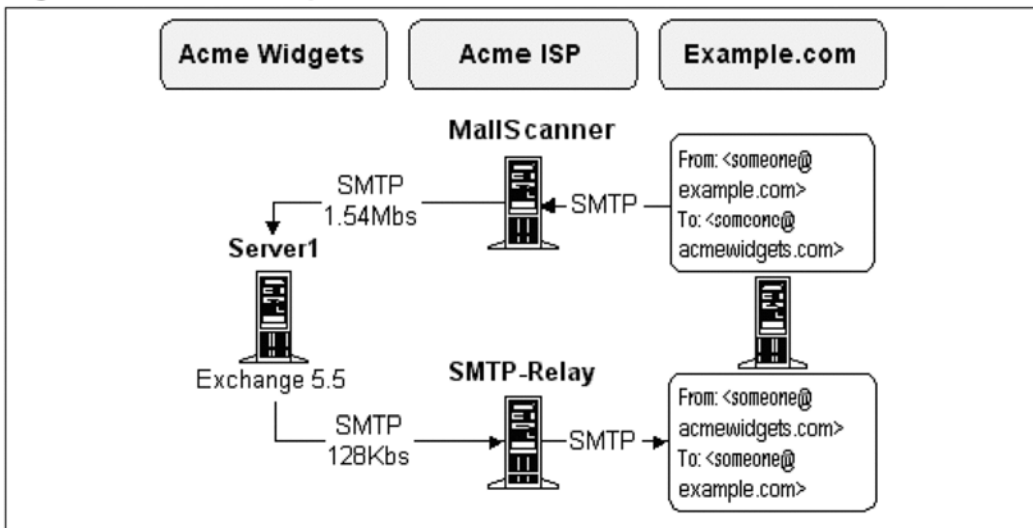
When designing e-mail infrastructure, it is often best to begin by determining the flow of e-mail to and from the Internet. To better illustrate this process, we will examine the Internet e-mail architectures of Acme Widgets and Ballystyx Engineering.

Acme Widgets accesses the Internet via a DSL (Digital Subscriber Line) connection. The connection provides high-speed (near-T1) download speeds, but only provides 128Kb upload speed. That means that receiving information from the Internet happens rapidly, but uploading information (including sending e-

mail) is considerably slower. To improve the performance of e-mail (especially when sending large e-mails to multiple recipients), Acme ISP set up an SMTP relay machine. In order to limit outbound spam and viruses, Acme ISP does not allow DSL customers to send out e-mail directly to Internet mail servers. All e-mail must go through the relay.

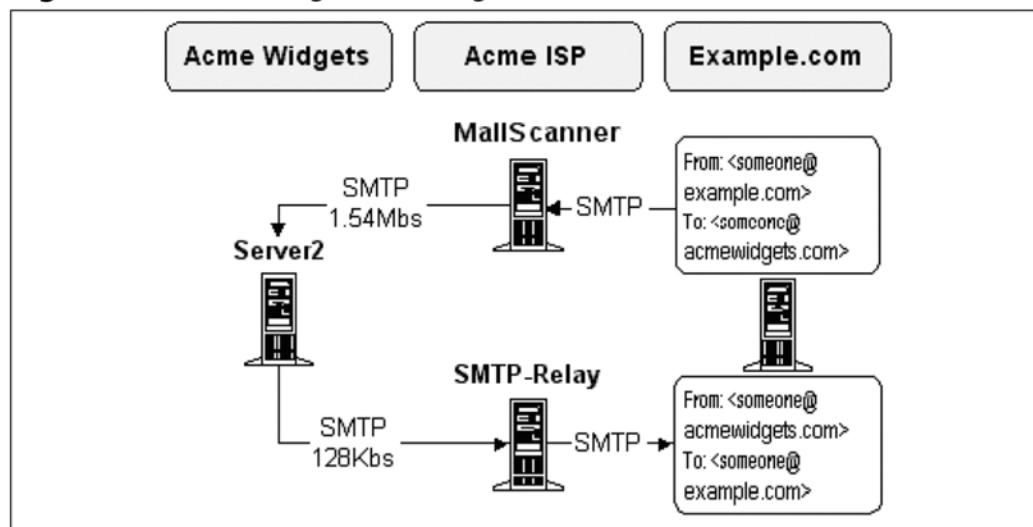
Acme ISP's commitment to fighting spam and viruses is not limited solely to outbound e-mail. Acme ISP manages the incoming e-mail for AcmeWidgets.com, and uses MailScanner to remove spam and viruses before the e-mail is received by Acme Widgets' Exchange server. The MX record for acmewidgets.com points to MailScanner.AcmeISP.net. Figure 7.7 illustrates Acme Widgets' current Internet e-mail architecture.

Figure 7.7 Acme Widgets Current Internet E-mail Architecture



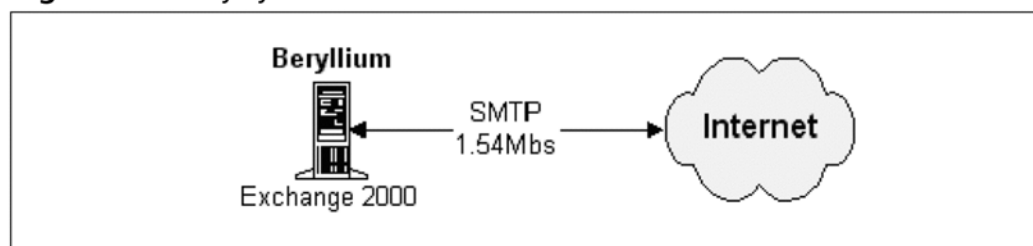
Because of the Internet e-mail architecture constraints imposed by the Internet Service Provider (ISP), there can be only one e-mail flow design for Acme Widgets. After Acme Widgets migrates to Linux, the e-mail architecture and flow will look like Figure 7.8

Figure 7.8 Acme Widgets Post-Migration Internet E-mail Architecture



Ballystyx Engineering accesses the Internet via a T-1 connection, which provides for high-speed download and upload of data. Although the T-1 provides significant bandwidth, it is somewhat clogged with spam and viruses. Because the ISP used by Ballystyx does not filter incoming e-mail or limit outbound e-mail, Ballystyx sends and receives e-mail directly to and from Internet e-mail servers. The MX record for ballystyx.com points to beryllium.ballystyx.com. Figure 7.9 illustrates the current Internet e-mail architecture and e-mail flow for Ballystyx Engineering.

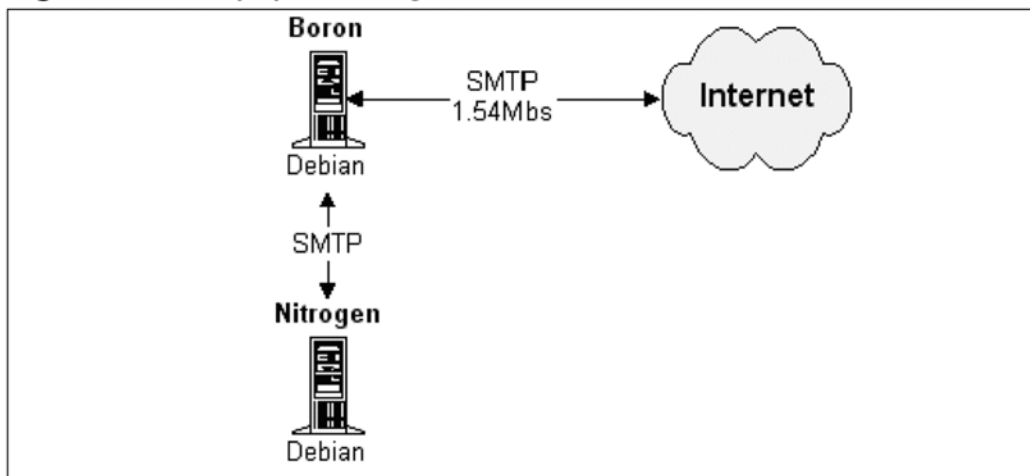
Figure 7.9 Ballystyx Current Internet E-mail Architecture



Vijay plans to deploy two new Linux servers, Boron and Nitrogen, as part of the Windows to Linux migration. Boron will replace Ballystyx's Exchange server with Linux-based messaging software, and will manage sending and receiving e-mail to and from the Internet. Nitrogen will be deployed to Ballystyx's Bangalore

office when there is sufficient time and budget. Figure 7.10 illustrates the post-migration Internet e-mail architecture for Ballystyx Engineering.

Figure 7.10 Ballystyx Post-Migration E-mail Architecture



Choosing Messaging Server Software

Now that the Internet e-mail architecture has been determined, we will focus on the software that will accomplish the functionality identified in the messaging requirements documentation. There are many criteria that can affect an organization's choice of software, including:

- IT staff's knowledge of applications
- Scaling and performance considerations
- Webmail requirements
- Compatibility with existing (legacy) applications
- Application feature set and capabilities
- Integration requirements
- E-mail client constraints

There is a considerable variety of open source messaging server software available for Linux. While there is not enough space to list all of the software available, Table 7.6 lists a number of open source messaging system components

recommended for Windows to Linux migration. More information about these applications may be found at their respective websites.

Table 7.6 Recommended Open Source Messaging Server Applications

MUA	SqWebMail, SquirrelMail, OpenWebmail
MTA	Courier-MTA, postfix, exim
MAA	Courier-IMAP, Cyrus-IMAP, gnupop3d
MDA	maildrop, procmail, or any other MDA

Because of the modular nature of Linux messaging systems and adherence to open standards, any combination of MUAs, MTAs, MAAs, and MDAs may theoretically be “glued” together. In practice, the configuration process is much easier when the various components are packaged together in an integrated suite, as is the case with Courier.

It is important to note that most Linux distribution vendors customize the packages for these messaging server applications to conform to the environment of the distribution. In many cases packages just work upon installation, or after answering a few configuration questions. The work put into application packaging by the Linux vendor can significantly lessen the effort required by a system administrator.

For a small company like Acme Widgets with a single server and minimal requirements, Sam chose the Courier suite. This provided an easy-to-deploy, integrated solution that met, and in some areas exceeded, the messaging needs of Acme Widgets. Sam found the Courier solution attractive because Courier could replace all of Exchange’s relevant features and offer an easy migration solution requiring minimal time to get configured and working. With the use of the migration scripts, migrating all of Acme Widget’s e-mail data to the new Courier mail server required less than an hour of work.

Because Ballystyx has more complex messaging requirements, Vijay chose a messaging suite that is composed of multiple best-of-breed open source applications. Although the Courier suite was attractive, Vijay and his staff already had experience in postfix administration, and liked the features and layout of the SquirrelMail webmail client better than the Courier SqWebMail client. Vijay decided that Courier-IMAP would be the preferred MAA for Ballystyx.

Determining How to Store the E-mail

One of the most important decisions in designing an e-mail system is where the e-mail will be stored. Will it be stored centrally, on the e-mail server, or will it be stored on the local hard drives of the clients? There are many factors that help shape such a decision including:

- Are there laptops that require e-mail operation while disconnected from the corporate network?
- Do users need the ability to access their mailbox from more than one computer within the organization?
- Will remote access (via the Internet, for example) to mailboxes be supported? Which protocols will be supported remotely?
- Will webmail access be required for some or all of the users?
- Does IT wish to support one protocol for mail access, or two?
- Do IT support personnel have time to service desktop / laptop users who download all of their e-mail to a POP client that “a friend helped configure”? In these cases the e-mail needs to be uploaded back to the mail server using IMAP, in a process that is somewhat complex and often time consuming.

Local storage of e-mail makes it more difficult to access the mailstore from multiple locations, and makes centralized backup of e-mail more challenging. If the organization wishes to make webmail available to all users, e-mail must be stored on a server. In addition, IMAP works best when stored on a server.

Although each organization will have varying needs and must tailor the solution to fit the specific requirement, most organizations that offer enterprise-class e-mail services (including shared folders) will make use of server-based e-mail storage and utilize IMAP as the MAA protocol.

Designing & Planning

E-Mail Backup and Restore

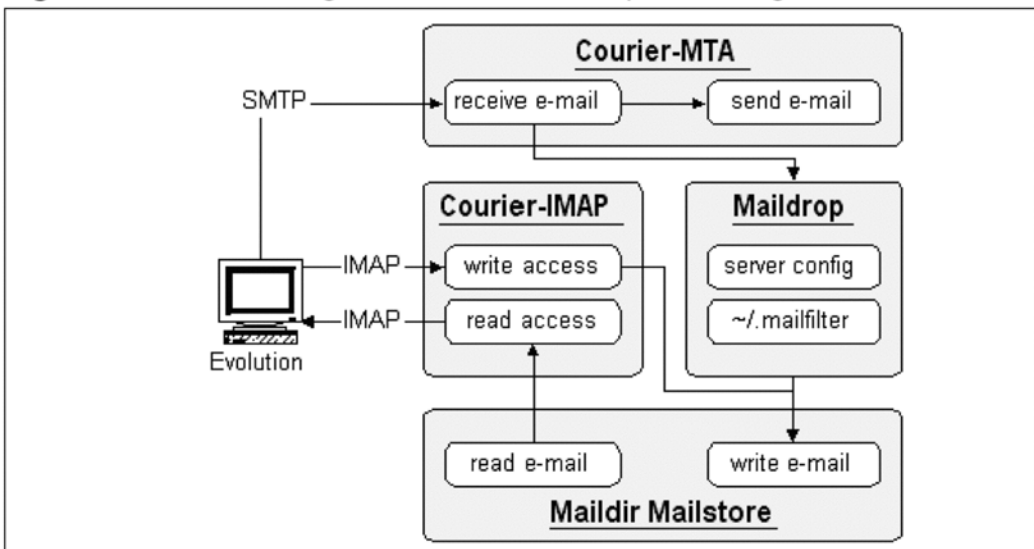
For Linux mailstores that use file-based storage (maildir, mbox), backup and restore is generally as simple as copying the files and directories to tape, then restoring specific files and/or directories if an e-mail gets accidentally deleted or the e-mail server's storage system fails.

Creating a Linux E-mail Server Diagram

Now that we've determined the details of the messaging services design (aside from anti-spam and anti-virus), we create a diagram that illustrates the design decisions for each of the messaging system components. This diagram is similar to the conceptual diagram in the "Understanding Linux-Based Messaging Services" section, but the names of the messaging system components have been replaced with the names of the software applications providing those services.

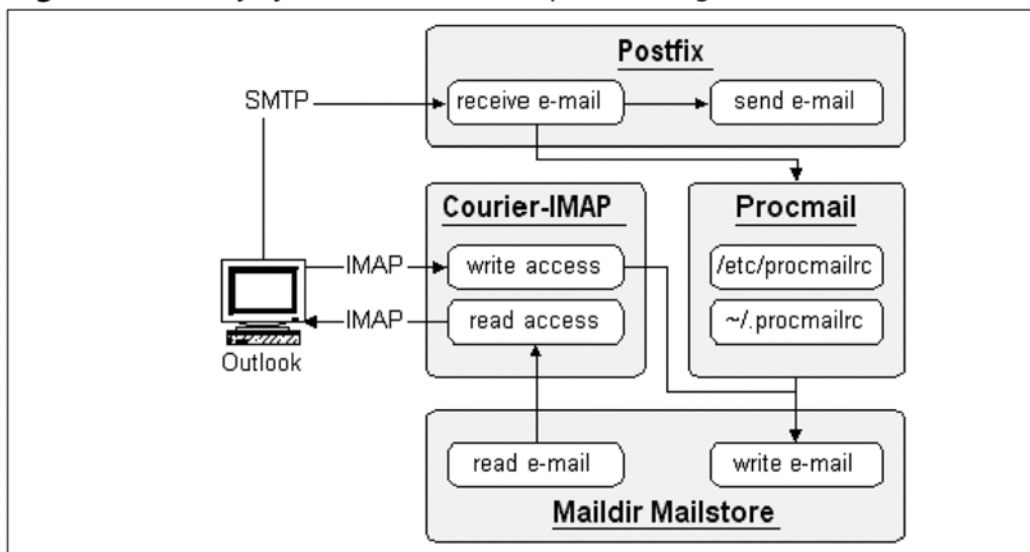
Acme Widgets is using the Courier suite to provide all of its messaging services. Figure 7.11 illustrates the design of the e-mail server components for Server2.

Figure 7.11 Acme Widgets E-mail Server Component Diagram



Ballystyx Engineering is using mail server components from multiple open source projects. Figure 7.12 illustrates the design of the e-mail components for Boron and Nitrogen.

Figure 7.12 Ballystyx E-mail Server Component Diagram



Integrating Anti-Spam and Anti-Virus Services

Although Internet e-mail has brought the benefit of rapid global communication, it has also brought on new sets of problems. As most of us have experienced firsthand, two of these problems are spam and viruses.

For most modern companies with a registered Internet domain, anti-spam and anti-virus services aren't an option – they're a requirement. The cost of productivity loss from spam and viruses for a medium-sized or larger company is far greater than the cost of providing anti-spam and anti-virus services. When the anti-spam and anti-virus software is open source, the cost is negligible to protect Windows computers from e-mail-borne infection and save the staff from spending time reading and deleting spam.

The following sections contain an overview of spam, viruses, and open source anti-spam / anti-virus software. The final section teaches how to design anti-spam and anti-virus services that are integrated with your messaging system. The

benefits and disadvantages of multi-server, single-server, and outsourced architecture are discussed.

Understanding Spam

Spam refers to unsolicited commercial e-mail, which generally means nearly identical e-mail that is bulk-delivered and therefore spread around like Spam (think canned ham). These e-mails are typically sales pitches to increase or decrease the size of various parts of the human body, earn money / decrease debt quickly, or obtain some other “desirable” product or service. Although the “click-through” rate of gullible people is typically less than 1%, it costs so little to send spam that even with such a low “hit” rate, spam is profitable to both the spammers and the advertised businesses. In 2004 (the publication year of this book), more than half of the e-mail traffic traversing the Internet is spam, and the percentage continues to increase, year after year.

A number of solutions have been tried to curb spam, including SPF (Sender Policy Framework). SPF works by using a DNS record that indicates which computers are permitted to send out e-mail on behalf of that domain. Although SPF has not yet been widely adopted at the time of this writing, it has been adopted by some prominent e-mail domains, including earthlink.net, gmail.com, mail.com, and spamassassin.org.

Some governments have passed laws (such as the U.S. CAN-SPAM act) to restrict sending unsolicited advertisements. Due to the difficulty and expense involved with tracking down and prosecuting offenders, such laws have limited effectiveness, or simply push the activity offshore to countries with less restrictive laws or enforcement. Because spam continues to be somewhat profitable, the onslaught is expected to continue for the foreseeable future. Fortunately, there are effective open source tools that block nearly all spam, with low false positive and false negative rates.

Determining if an E-Mail is Spam

The first step to stopping spam from entering your network is to determine if an e-mail is spam or *ham* (good e-mail). There are a number of approaches to detecting spam, and an effective anti-spam application will make use of as many of these techniques as is feasible. It is important to note that each additional test increases the processing overhead for each message. Tests to determine if an e-mail is spam include:

- Spam *fingerprint* matching
- Message header analysis
- Message body analysis
- Rules-based (heuristic) analysis
- Bayesian analysis

Most anti-spam systems assign each e-mail a score that indicates the likelihood that the message is spam. For SpamAssassin, the scale ranges from zero (about 0% chance of spam) to 10 or beyond (nearly 100% likely spam). In well-designed spam scoring systems, there is no single factor that determines that an e-mail is spam. Instead, the various weights of multiple factors add up to a spam score that is greater than a threshold set by the e-mail administrator.

In most anti-spam systems, the spam score of an e-mail is the criteria that determines what action(s) will be taken by the anti-spam software. Messages that are almost certainly spam are simply deleted without delivery. Messages that are likely to be spam, but not quite so certainly, may be placed in a separate queue to be reviewed by an e-mail administrator and determined to be spam or not spam. Such review can be valuable to teach the anti-spam system what is and isn't spam, and will help the anti-spam system to score these types of e-mails more accurately in the future.

Another way to deal with spam is to mark (and forward) e-mails that are likely spam, typically by prefixing the subject line with something resembling **★★SPAM★★**. In these cases, all of the e-mail is delivered; none are dropped. This has the advantage of never losing an e-mail message because of a false positive spam score, but does require increased storage space and/or additional effort from the mailbox owner. In most cases, it is best to stop e-mails with a very high spam score at the perimeter e-mail device, and not even waste storage space or processing time on spam.

Spam fingerprinting is one of the most effective ways of determining if an e-mail is spam. This method is a cooperative effort that relies on the premise that many spam e-mails are nearly identical. When a subscriber receives a new e-mail, a *fingerprint* of the e-mail is computed. Then the fingerprint is matched against the spam fingerprint database. If there is a match, it means that another subscriber has already reported a nearly identical e-mail as spam. If there is no spam fingerprint match and this e-mail is later determined to be spam, the subscriber may upload the fingerprint of the new spam to the spam fingerprinting database.

DCC, Razor, and Pyzor are three open source anti-spam organizations that act as centralized reporting and distribution servers for spam fingerprints and checksums. You may use one, two, or all three of these services to increase the effectiveness of your anti-spam operation. Table 7.7 lists the full names and websites.

Table 7.7 Spam Fingerprinting Organizations

Full Name	Website
Distributed Checksum Clearinghouse	http://www.rhyolite.com/anti-spam/dcc/
Pyzor	http://www.pyzor.sourceforge.net/
Vipul's Razor	http://www.razor.sourceforge.net

In addition to spam fingerprinting, there are a number of heuristic factors that contribute to (raising) the spam score of an e-mail. For starters, if the e-mail message is being sent from an IP address that is a known spammer, the spam score will be raised significantly. If the e-mail subject or body text contains certain words, such as “Viagra”, or a gapped or disguised version of the word, (such as “V i.ag r_a”) the spam score will increase. If there are problems with the format of the e-mail, including invalid MIME encoding, bogus header lines, or other characteristics that violate RFCs, the spam score increases significantly. If the body of an e-mail is mostly a group of HTML pictures, or contains obfuscated or invalid HTML, the spam score is raised. If the sender or recipient(s) contain odd combinations of letters and numbers as the e-mail address(es), or if the sender’s domain name is yahoo.com and the e-mail is being sent by an IP address from the Earthlink.net dial-up pool, the spam score is increased. And finally, if the e-mail contains a “remove from list” URL that claims to honor all removal requests, the spam score increases significantly – this is a favorite trick of spammers to verify that an e-mail address connects to a live user, and increase the flow of spam to that e-mail address!

Bayesian analysis is another effective spam-fighting technique. This method relies on a database of words or phrases (called *tokens*) that appear frequently in ham, but almost never in spam, as well as a second database of tokens that appear frequently in spam, but almost never in ham. A Bayesian analysis spam filter is unusual because it must learn about spam and ham before it is useful. In general, the greater the quantity of known spam and ham that is analyzed by a Bayesian engine, the more effective it is at fighting spam.

Fighting spam is difficult because the nature of spam keeps changing. The continual increase in the complexity of spam and anti-spam technology is similar to an arms race, with both sides trying to outmaneuver each other. Fortunately, an army of open source programmers and e-mail administrators help to keep spam under control by responding to new spam, updating heuristic filters, and reporting spam to spam fingerprinting organizations.

Whitelisting and Blacklisting

Whitelisting and blacklisting refer to methodologies employing lists that are used to grant or deny access. When these terms are applied to spam, this means that an e-mail may be classified or declassified as spam based on the sender's IP address and/or e-mail address/domain. Whitelisting and blacklisting can be very effective, and can be used to avoid false positives and false negatives in spam detection. Whitelisting can help when people are subscribed to e-mail lists, receive opt-in marketing e-mail, or have occasional problems with a well-known sender/recipient's e-mail not getting through a spam filter.

One of the most effective ways to use blacklisting to fight spam is to utilize the MAPS RBL (Realtime Blackhole List) service. The RBL contains a carefully-maintained list of the IP addresses that are known to send spam. More information about MAPS RBL is at www.mail-abuse.com/services/mds_rbl.html.

Although there is some administrative overhead involved with their use, blacklists and whitelists are recommended tools to increase the effectiveness of anti-spam operations.

Understanding Viruses

In addition to spam, the other problem that arrived with widespread use of e-mail and Windows computers is e-mail-borne viruses. E-mail-borne virus infection methodologies typically entice the user to view the virus-bearing e-mail message (or attachment) with a suggestive (free prize, exciting picture, etc.) subject. These types of attacks are called *Trojan horse* attacks, named after the wooden gift horse that allowed the Greeks to enter and capture the city of Troy. When we refer to viruses in this book, we are referring to viruses, worms, and trojans.

The infection vector of e-mail-borne viruses fall into two categories:

- Viruses that infect by taking advantage of flaws in commonly used e-mail clients, typically the HTML rendering portion of those clients. The

most frequent examples of this are Outlook/Express and IE HTML rendering engine exploits (MSHTML.DLL).

- Viruses that infect by masquerading as a safe attachment. When the user opens the trojan attachment, a virus script or other type of malicious file is launched.

The payload of e-mail-borne viruses ranges from bad to worse. Table 7.8 lists the common types of payloads found in e-mail-borne viruses.

Table 7.8 Types of Virus Payloads

Method	Description	Effect
E-mail Replication	The e-mail virus mails out copies of itself to users in an Exchange Global Address List, a personal address book, or e-mail addresses harvested from files on the hard drive. When users open the infected e-mail or attachment, they mail out more copies of the infected e-mail.	Overwhelms MTAs and devours all available storage space in the mail store.
Network Replication	The virus spreads by connecting to other computers and exploiting a security flaw to infect the target host, or places copies of itself in files on network shares. The newly infected host then attempts to contact and infect other hosts.	Can overwhelm network bandwidth.
Annoying	The virus may display any annoying message, but doesn't usually attack other machines.	Annoying, but not damaging.
Backdoor	The e-mail virus installs software that lets a hacker remotely control the computer. The virus may routinely contact a server that provides updated instructions.	These hacker-controlled computers may later launch attacks and cause serious disruptions.

Continued

Table 7.8 Types of Virus Payloads

Method	Description	Effect
Keystroke Logging	The virus logs all keystrokes that the user types, including usernames and passwords. The information may be stored locally or uploaded to a hacker’s computer.	Hackers can learn valid usernames and passwords, and use these to launch attacks.
Destructive	The payload executes an attack on the local machine and attached file shares that may erase or alter files. The virus may attack other networked machines.	Similar to a standard desktop virus outbreak, the cleanup usually involves restoring backup files from tape.
Hybrid	A hybrid virus makes use of two or more types of payloads and/or infection routes, and may also incorporate advanced features such as polymorphism to avoid detection by anti-virus software.	Can be quite destructive and difficult to quarantine/inoculate.

Certain types of files spread viruses more easily than others. The file has to be of a type that can be executed by the underlying operating systems (usually Windows), or the file must contain an exploitable provision for scripting or macro usage.

While comprehensive virus control solutions must address containment and inoculation, the most desirable anti-virus solution is to stop the viruses from even entering into the organization. One of the easiest ways to protect against these types of viruses is to scan and/or block some or all of the attachments that could potentially contain viruses. Table 7.9 shows a list of file types that may be used to execute malicious code on a Windows computer. These types of files do not represent the same threat to Linux desktops.

Table 7.9 Attachment Block List

Extension	File Type
.PIF	Program Information File
.BAT	Batch file
.CMD	Batch file
.COM	Executable file
.EXE	Executable file
.VBS	Visual Basic Script
.SCR	Windows screen saver

It is worth noting that over 90% of e-mail-borne virus infections involve Outlook/Express and IE (usually MSHTML.DLL) flaws. Simply eliminating the use of Exchange, Outlook/Express, and Internet Explorer will go a long way toward preventing these types of attacks from succeeding. Switching to Linux-based messaging and Mozilla will achieve a security improvement with regards to viruses, and migrating Windows desktops to Linux will achieve an even greater security improvement.

Exploring Open Source Anti-Spam / Anti-Virus Applications

Now that you understand the theories behind anti-spam and anti-virus services, let's take a look at the open source applications that actually accomplish these important tasks. In this section we will examine three of the most popular open source anti-spam and anti-virus applications:

- SpamAssassin
- Clam AntiVirus
- MailScanner

Although there are a number of open source applications that perform anti-spam and/or anti-virus functions, these three applications are widely used for anti-spam and anti-virus services, and provide an easy-to-deploy solution that meets the needs of nearly all organizations.

SpamAssassin

SpamAssassin is a very popular open source anti-spam application. In this book, when we refer to SpamAssassin, we are referring to SpamAssassin 3.0.x.

Originally a Deersoft product that was open-sourced, SpamAssassin is a set of Perl scripts that perform heuristic analysis on e-mail headers and body text. These procedures include word and phrase matching, obfuscation detection, HTML content analysis, e-mail header and address analysis, Bayesian analysis, spam fingerprint matching, and blacklist/whitelist operations. In short, SpamAssassin is a mature product that has virtually every anti-spam feature needed by an organization.

SpamAssassin is easily integrated with Postfix, Exim, Sendmail, or any other MTA and most open source messaging software. Many open source (and even some commercial) products make use of SpamAssassin for their anti-spam functionality. SpamAssassin is used by many ISPs to provide low-cost anti-spam services to their customers. Configuring SpamAssassin is easy using Webmin (www.webmin.com) or the quick-configurator at www.yrex.com/spam/spam-config.php

SpamAssassin can make use of all three of the no-cost spam fingerprinting databases. Documentation about integrating these services with SpamAssassin is at the following URLs.

- <http://wiki.apache.org/spamassassin/UsingDcc>
- <http://wiki.apache.org/spamassassin/UsingPyzor>
- <http://wiki.apache.org/spamassassin/UsingRazor>
- <http://spamassassin.apache.org>

Clam AntiVirus

ClamAV is a popular command-line GPL (General Public License) virus scanner. In this book, when we refer to ClamAV, we are referring to ClamAV version 0.80. ClamAV detects and isolates over 20,000 viruses, and integrates with virtually any MTA or mailstore. It even detects viruses in zip, gzip, bzip2, rar, and other file archives. ClamAV includes an automatic database updater (freshclam) with support for digital signature verification.

ClamAV is used by many ISPs and by high-profile open source domains such as SourceForge.net and RoaringPenguin.com. The ClamAV virus database is

updated several times per week, and the ClamAV virusdb team responds quickly to new virus threats.

More information about ClamAV is available at www.clamav.net.

MailScanner

MailScanner is an ingenious product that combines anti-spam, anti-virus, and other malware filtering into a single package that works with SpamAssassin, ClamAV, and other virus scanners. MailScanner is a well-known, mature, widely used integrated solution that works with virtually any open source MTA, anti-spam, or anti-virus software. For most businesses, MailScanner is an excellent choice in terms of feature set and ease of deployment. MailScanner scans e-mail for:

- Viruses, using up to 14 virus scanners, including ClamAV.
- Spam, using SpamAssassin, with a typical detection rate of 95% or greater.
- Other e-mail-based attacks and malware that takes advantage of known (and sometimes unknown!) security vulnerabilities.

In addition, MailScanner is configured in a defensive way to prevent denial of service and resource exhaustion attacks. It will remove malicious objects when possible, and quarantine sections when necessary. MailScanner is used by thousands of organizations and hundreds of thousands of users protecting millions of messages every day. MailScanner is highly configurable, but is easy to install and set up with popular MTAs or in gateway mode. How-to's are listed on the website that explain (step-by-step) how to glue MailScanner to a mail server, although the simplest way to accomplish this is by using MailScanner in gateway mode.

For more information about MailScanner, visit the website at www.mailscanner.info.

Designing Anti-Spam / Anti-Virus Services

Now that you fully understand spam, viruses, and open source anti-spam / anti-virus applications, you are ready to design an anti-spam / anti-virus solution that meets or exceeds the requirements of your organization. This book addresses four types of architecture solutions for anti-spam / anti-virus services: e-mail gateway, shared-server, commercial appliance, and outsourced service.

For a very small company like Acme Widgets with limited budget and IT expertise, the simplest solution was using the free anti-spam and anti-virus ser-

vices provided with the Acme ISP domain services contract for AcmeWidgets.com. In this outsourced solution, Acme Widgets gets all of the benefits of anti-spam and anti-virus services without the disadvantages of dealing with server or software management.

For Ballystyx Engineering, Vijay chose a dedicated anti-spam / anti-virus e-mail gateway solution. This has the advantage of segregating anti-spam / anti-virus services to a single-purpose server, and easily allows Vijay to upgrade Ballystyx's anti-spam / anti-virus infrastructure without affecting internal e-mail service.

Shared E-Mail / Anti-Spam / Anti-Virus Server Option

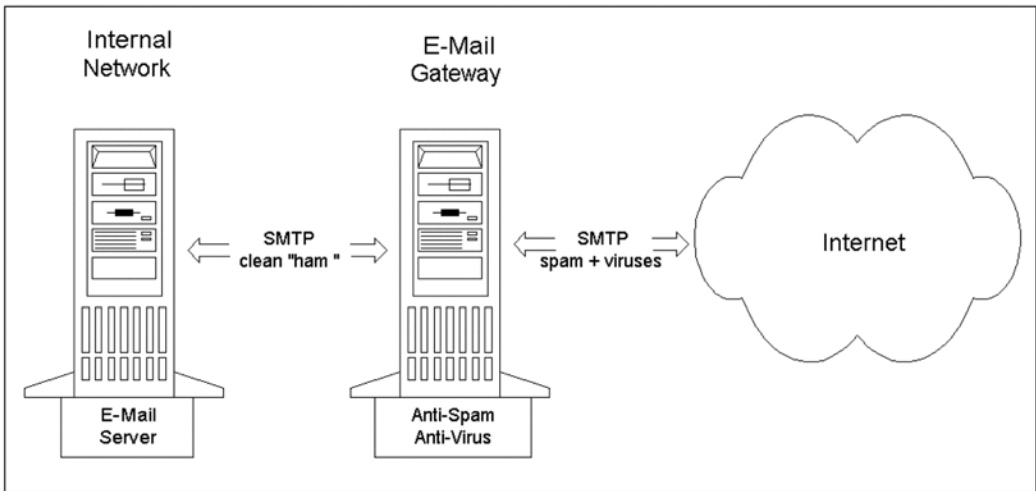
Installing all e-mail, anti-spam, and anti-virus software on the same machine is a widely used option, especially in circumstances with limited budget. This usually means integrating SpamAssassin, ClamAV, and/or MailScanner with the server's MTA. The configuration intricacies of such an installation are beyond the scope of this book. For full coverage of this topic, consult the MailScanner documentation at www.mailscanner.info.

This solution has the benefit of being highly configurable and being able to reject many types of spam or malware prior to entering a user's mailbox (but not prior to entering the corporate network). This can be one of the most time-consuming solutions to set up, configure, troubleshoot, and tune, although it does allow for maximum control over the anti-spam/virus/malware environment. If you are considering this solution, keep in mind that running all of these processes on a single machine can tax the computer's resources, and that a shared-server solution may face significant scalability (or tuning) challenges as the number of users increases. Many companies start with a shared-server solution when small, but later grow into the dedicated e-mail gateway solution described below.

Dedicated Anti-Spam / Anti-Virus Gateway Option

Another approach to spam and virus control is to utilize a computer that is dedicated to anti-spam / anti-virus e-mail processing. This e-mail gateway computer receives SMTP connections from the Internet and performs all spam/virus/malware checks prior to allowing an e-mail to enter the network of internal e-mail servers. Figure 7.13 illustrates this design.

Figure 7.13 Anti-Spam / Anti-Virus E-mail Gateway



From a number of perspectives, this is an attractive option. Maintenance, integration, and scaling considerations are segregated to another machine, and performance issues with the e-mail gateway won't diminish the performance of the e-mail server for user mailboxes. Performance issues with the gateway may delay e-mail delivery, but will not affect the e-mail experience for the users. If there is a budget, this is usually a great way to place the anti-malware functions on a separate machine and ensure horsepower is available for those functions – even as e-mail volume continues to grow.

Ballystyx liked this idea so much that Vijay decided to go one step further. Because Ballystyx's ISP had an inventory glut of inexpensive co-lo servers, Vijay was able to lease a used server at a very low monthly cost instead of purchasing a new server. Using this solution with MailScanner, Vijay is able to provide anti-virus and anti-spam services to Ballystyx while freeing up Internet connection bandwidth that had been previously consumed by viruses, spam, and other malware.

Even though the machine (Mercury) is physically located at the ISP instead of within Ballystyx, this posed no issue for Vijay because he could use ssh (access granted only from Ballystyx's IP range) to securely manage Mercury. The facilities at the ISP were more advanced than at Ballystyx, which meant that Mercury could take on a longer-term power outage and had better earthquake protection. The mail gateway was safer at the ISP, and could continue collecting and processing e-mail even if Ballystyx's own computer room suffered an outage. This solution was advantageous for Ballystyx in many ways including increasing the speed of all Internet services by decreasing bandwidth used by e-mail.

If legal requirements require the archiving or filtering of types of e-mail messages, the e-mail gateway may be configured to process outgoing mail, or a second server may be added to facilitate this goal. This function may also be handled by any of the MTAs mentioned in this book.

Commercial Appliance Option

In some cases, it might be simplest to purchase a low-cost anti-spam / anti-virus e-mail gateway appliance. The appliance may be deployed at the ISP or the computer room of the company's headquarters. There are numerous vendors offering a considerable variety of devices appealing to the needs of any size or type of company. Some of these e-mail gateway appliances even run open source software. Nearly all of these devices feature a web-based configuration interface for easy management of the device.

These appliances are typically sold with a subscription and support agreement. This agreement usually includes a mechanism for receiving software, rules, and signature updates. Keep in mind that updates will usually cease if the agreement is not paid or renewed, so you will be locked into an agreement with the vendor unless the appliance uses open source software that you can manage yourself.

Even with the lock-in potential, this solution has the advantage of freeing your organization from the management and administrative requirements associated with providing anti-spam and anti-virus services. In some organizations, particularly those with limited knowledge of (or desire to learn) anti-spam / anti-virus technology, this can be an appealing solution.

Outsourced Option

Although the multi-server and commercial appliance solutions are suitable for businesses with appropriate budget, time, and staff, opting for anti-spam and anti-virus services that are provided by an ISP (as in the case of Acme Widgets) saves a company the trouble (and any potential headaches) of managing these services. If reasonably priced, it can also be the least expensive choice, and is frequently featured as part of a domain-hosting package.

For companies that wish to avoid the hassle of e-mail management altogether, there is the possibility of outsourcing e-mail completely to an ISP, in which case the ISP manages the Internet domain (incl MX records) e-mail addresses, incoming (and outgoing) mail services (SMTP), and mailbox storage and access (POP or IMAP). Although this almost completely removes the man-

agement overhead of hosting Internet domain e-mail services, it also limits flexibility, and may require the downloading of e-mail to local users' workstations, which limits the availability of services such as webmail and e-mail that are accessible via any workstation within the company. This client-downloaded e-mail also introduces additional backup complexities and concerns.



Migrating Information from Exchange to Linux

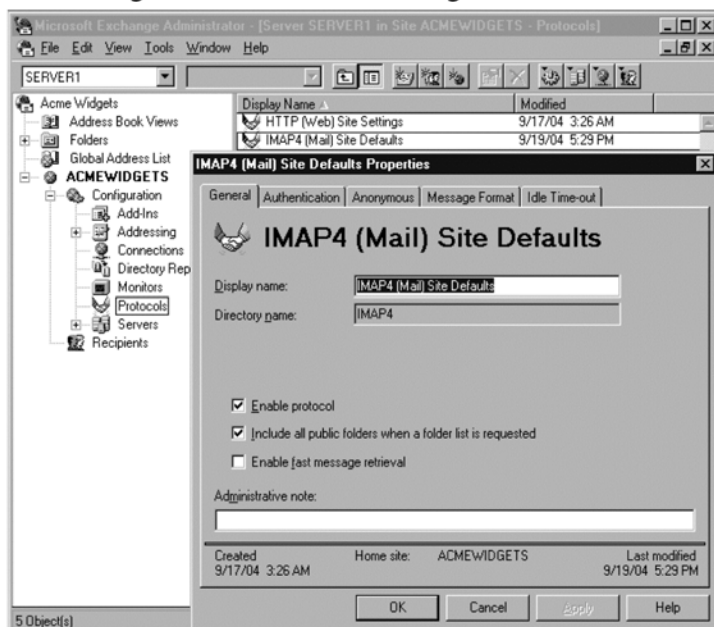
This migration section will focus on migrating e-mail in mailboxes stored on Exchange servers to a running Courier IMAP instance. The script associated with this migration example was written for Courier, although other source and target IMAP servers may also work. If you are in a situation where all e-mail is downloaded from the Exchange server to a local e-mail client (such as Outlook or Mozilla), read Chapter 11, “Inside the Linux Desktop,” for information on migration of e-mail. This migration will also only concentrate on mail objects and will not deal with contact, tasks, journal, and public folders. These Groupware features of Exchange are dealt with in the next chapter, “Groupware and Calendaring Services.” The scripts will need to have the Net::LDAP and the Mail::IMAPClient perl modules installed as well

Preparing Exchange for Migration

In order to prepare Exchange 5.5 or Exchange 2000 for the style of migration provided by the scripts, there are a number of steps you must take. Since the e-mail migration scripts required IMAP services, you must ensure that IMAP services are enabled on the Exchange server(s):

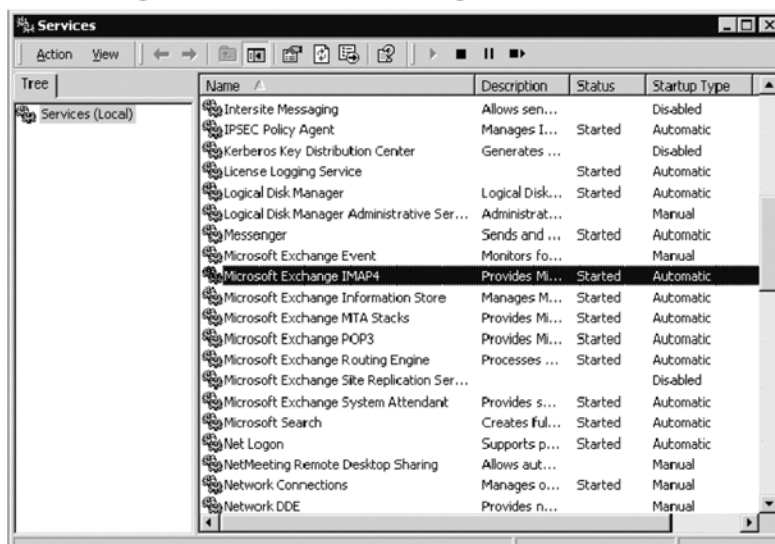
- Enable IMAP protocol mailbox server
- Create a .csv file containing user mailboxes and passwords for the users you wish to migrate. This is necessary since Exchange does not allow administrative access to mailboxes via IMAP. The file should contain a <username>, <password> set on each line.

Figure 7.14 Enabling IMAP Services (Exchange 5.5)



A normal installation of Exchange 2000 installs the IMAP4 service by default. You need to simply go to the control panel and make sure the IMAP4 service has been started. Figure 7.15 illustrates this procedure.

Figure 7.15 Enabling IMAP Services (Exchange 2000)




Installing the Courier IMAP Server Suite

Courier IMAP is a large suite of messaging services and it is recommended that you install the appropriate precompiled package for your Linux distribution. In the case of Fedora or RedHat AS 3.0, there is no default RPM, however a .spec file is provided in the Courier imap source tarball so that you can build packages for Redhat systems. Here's how you would do this:


- Download the latest courier tarball (0.47 as of this writing)
- Gunzip/untar the source to a temporary directory and copy the courier.spec to /usr/src/redhat/SPECS
- Copy the original tarball to /usr/src/redhat/SOURCES
- As root you may need to `chmod -R 777 /usr/src/redhat` to make sure that a non-root account can create RPM files here, since courier will not compile under the root account
- As a non-root user, change to /usr/src/redhat/SPECS and type `rpm-build -bb courier.spec`
- This will take a while to compile, but should result in several RPMs being built for your version of Redhat, in /usr/src/redhat/RPMS/i386
 - courier-0.47-1.i386.rpm
 - courier-mysql-0.47-1.i386.rpm
 - courier-fax-0.47-1.i386.rpm
 - courier-pgsql-0.47-1.i386.rpm
 - courier-imapd-0.47-1.i386.rpm
 - courier-pop3d-0.47-1.i386.rpm
 - courier-ldap-0.47-1.i386.rpm
 - courier-smtpauth-0.47-1.i386.rpm
 - courier-maildrop-0.47-1.i386.rpm
 - courier-webadmin-0.47-1.i386.rpm
 - courier-maildrop-wrapper-0.47-1.i386.rpm
 - courier-webmail-0.47-1.i386.rpm
 - courier-mlm-0.47-1.i386.rpm

- Install the appropriate packages for the functionality you desire, such as webclient and fax
- In most cases, you will not need to install ldap, postgresql, mysql, etc authentication modules since courier works well with PAM, making that the best choice




Now that you have Courier installed, you will need to configure this package. All of the options available are beyond the scope of this book, however please refer to the documentation in order to setup for your particular organization. Examples for Acme Widgets and Ballystyx are included on the companion CD for your reference.

Migrating E-mail to Linux



The script `w2lmt-migrate-imap` provided will automate the process of migrating private folders of email objects from MS Exchange to a running Courier IMAP server. Since Courier requires the use of Maildirs (covered previously in the chapter), and additionally Courier does not create these folders for individual users by default, the migration script will accomplish this automatically, however this requires you to run the script on a linux server that has write access to the user's home directory. The script is designed to work best with empty maildirs, however will attempt to use an existing instance if they exist. Here is how the script basically works, using Acme Widgets as an example:

```
w2lmt-migrate-imap -f migrate-imap-acmewidgets.conf
```

- 
- Check to make sure it has everything it needs to continue
 - Prompt for the LDAP administrator password
 - Read in the list of mailbox user/passwords from the csv file provided in the configuration script
 - For each user:
 - Read in the user's home directory from LDAP
 - Reset the user's Linux/Samba password to the match the exchange password
 - Create the proper Maildirs in the user's home directory
 - Connect to the Exchange server via IMAP



- Connect to the Courier IMAP server (should use the newly created Maildirs)
- Loop through the exchange mailbox and copy folders and mail message objects to the correct location on the Courier IMAP server. (Note, since Courier only allows folders to be created under the INBOX folder, all migrated folders and messages will appear as sub-folders under INBOX

It is important to note that as a result of this process, the user's password on the Linux/Samba PDC will now be set to match the password specified in the csv file. So be sure your users change their passwords for security as soon as you begin to use your new Samba PDC! Additionally, this script did not deal with Public folders. You will need to manually deal with these items since on Exchange these can contain various non-mail objects and special permissions.

Summary

After reading this chapter, you should have a good understanding of messaging in general, and Linux-based messaging in particular. You should know about MUAs, MAAs, MTAs, and MDAs, and understand how to design mail servers and messaging systems, including anti-spam and anti-virus.

If you've put this knowledge to good use, prototyped your messaging environment in the lab, tested it fully, deployed it to your users, and migrated their e-mail: Congratulations – you are no longer reliant upon Microsoft Exchange for messaging services. You now manage a messaging environment that's less resource-intensive, and can enjoy license-free messaging and anti-spam / anti-virus services.

With the completion of this chapter, you should have your messaging services functioning properly. This means that your users can connect IMAP (and/or POP) clients to retrieve their messages, and your organization can properly send and receive Internet e-mail. If you were using Exchange or another IMAP-capable MAA, you should have all of the important user mailbox e-mail transferred to the Linux-based mailstore.

If you've added open source anti-spam and anti-virus services to your messaging system, you've saved hundreds or thousands of dollars in licensing fees from Exchange-integrated anti-spam / anti-virus software. With this work complete, you can retire Microsoft Exchange unless you are migrating groupware / calendar information (covered in Chapter 8, "Groupware and Calendaring Services").

When you retire Microsoft Exchange, you frequently remove the last application that depends on Active Directory. In this case, you can also retire Active Directory and MS-DNS/DHCP services that were dependent on Active Directory, removing most of the MS infrastructure of an organization. For most organizations, this represents a core underpinning of the infrastructure, and a significant milestone in the Windows to Linux network services migration process.

Although designing, deploying, and migrating to Linux-based messaging services can require some effort, these services usually function reliably and require minimal maintenance over the long term.

Solutions Fast Track

Understanding Microsoft Messaging Services

- ☑ Exchange 5.5 / 2000 is Microsoft's messaging (and groupware) services solution for Window NT and Windows 2000 Server. Exchange offers shareable user mailboxes, e-mail groups, shared public folders, IMAP and POP interfaces, shared contacts, user and group calendars, flexible database storage, and almost any feature required of an enterprise messaging system.
- ☑ A typical Exchange 5.5 server is comprised of five subcomponents including a monitoring attendant, directory server, a mailstore, an MTA, and an SMTP gateway. Exchange 5.5 supports MAPI, IMAP, POP, and HTTP-accessible messaging and groupware services.
- ☑ Exchange 2000 offers the same (and/or upgraded) capabilities offered by Exchange 5.5. However, Exchange 2000 does not offer a directory service and instead requires Active Directory. When Exchange 2000 is installed, it updates the Active Directory schema to e-mail-enable the relevant objects.

Understanding Linux-Based Messaging Services

- ☑ Most Linux-based messaging systems are comprised of an integrated, yet modular system comprised of multiple component responsible for transmitting, receiving, delivering, and providing access to e-mails. These components include a mail transfer agent (MTA), a mail delivery agent (MDA), a mailstore, a mail access agent (MAA), and a mail user agent (MUA).
- ☑ The history of popular open source messaging server software began with Sendmail in 1980, followed by Qmail, Postfix, Exim, and Courier.
- ☑ Sendmail, Qmail, Postfix, Exim, and Courier are all very capable MTAs that enjoy widespread use. Each has strengths and weaknesses. Courier has the distinction of being an all-in-one solution that includes an MTA, Webmail MUA, MAA, MDA, calendaring / groupware features, mail list manager, and usage reporting tools.

Designing Linux-Based Messaging Services

- ☑ Designing Linux-based messaging services largely consists of determining the path for sending and receiving Internet e-mail, and determining the type, sizing, placement, and configuration of each messaging system component application. The primary goals of messaging system design should be to design a system that provides reliable functionality to the users and scales to meet the expected volume of e-mail for the next 2-3 years.
- ☑ Many factors can affect an organization's choice of messaging software, including the staff's (lack of) knowledge about specific applications, scaling considerations, compatibility with existing (legacy) applications, integration requirements, and e-mail client constraints.
- ☑ For a small company like Acme Widgets, Courier provides an easy-to-deploy solution that meets and in some cases, exceeds the messaging needs of Acme Widgets. For a larger company like Ballystyx Engineering, a popular choice is the combination of Postfix, Courier-IMAP, and SquirrelMail.
- ☑ Two important design considerations are where to store the e-mail and how to provide access. For most organizations, storing the e-mail on the server and providing IMAP access to mailboxes are the recommended choices.

Integrating Anti-Spam and Anti-Virus Services

- ☑ Although Internet e-mail has brought the benefits of rapid global communication, it has also brought the problems of spam and viruses. For any sizeable modern company with a registered Internet domain, anti-spam and anti-virus services aren't an option – they're a requirement.
- ☑ Spam refers to unsolicited commercial e-mail that is bulk-delivered to Internet recipients. Spam detection consists of spam *fingerprint* matching, message header and body analysis, heuristic analysis, and Bayesian analysis. Each e-mail is assigned a numerical score that indicates the likelihood that the message is spam.

- ☑ Viruses refer to viruses, worms, trojans, and other malware that infects and replicates. Virus payloads range from unpleasant to very destructive. Anti-virus strategies include blocking attachment types that may contain viruses, and scanning e-mail for virus signatures.
- ☑ SpamAssassin, ClamAV, and MailScanner are three recommended applications to add anti-spam and anti-virus services to a messaging system. MailScanner combines spam detection and virus scanning in an integrated package.
- ☑ There are four types of anti-spam / anti-virus architectures explored in this book: e-mail gateway, shared-server, commercial appliance, and outsourced service.

Migrating Information from Exchange to Linux

- ☑ The migration methodology employs an IMAP to IMAP mailbox copy
- ☑ Ensure that IMAP services are enabled on Exchange
- ☑ Create a csv file containing username, password for each mailbox to migrate
- ☑ Install and configure Courier IMAP server suite
- ☑ Run the migration script on a linux host with WRITE-ACCESS to each user's home directory

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: How do I set up a maildir storage?

A: Maildirmake may be used to configure a directory for maildir storage. Type **man maildirmake** for more information.

Q: Is there an easy way to migrate from mbox to maildir format?

A: Yes. The utilities `mbox2maildir` and `maildir2mbox` accomplish these tasks. Download the Qmail distribution and/or visit www.qmail.org for more information.

Q: After running the migration script, I connect to my new Courier IMAP mailbox, but there are no folders under INBOX!! Where is all my mail?

A: Depending on your mail client you may need to subscribe to the subfolders in order for them to appear properly.

Groupware and Calendaring Services

Solutions in this Chapter:

- Understanding Exchange and Outlook Groupware and Calendaring Features
- Understanding Linux-Based Groupware and Calendaring Services
- Migrating to Linux Groupware / Calendaring Services

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Groupware means many things to many organizations (and vendors). Groupware primarily means calendaring, and in the context of groupware, that means shared schedules. For some, it's just having a web-accessible way to view and edit one's schedule. For others it means having the ability to schedule a meeting with access to the free/busy times of all meeting attendees, sending out a graphically-formatted invitation, and allowing users on multiple platforms to accept/decline the meeting with updates provided to a calendar server.

Another form of groupware is a collaborative content management / development system such as TWiki and other WikiWikiWeb derivatives. Sourceforge is an additional example of a collaborative content development groupware environment. Sourceforge's CVS- and web-based software development and distribution capabilities enable it to host much of the world's open source software.

Groupware also means something else. Because groupware can mean any software that's related to collaboration, almost anything that has to do with electronic communication qualifies as groupware. In this context, e-mail, file sharing, voicemail, fax, and online conferencing are all forms of groupware.

Deploying Linux-based groupware usually requires a LAMP (Linux-Apache-MySQL-PHP/Perl/Python) environment. A detailed description of the LAMP framework is outside of the scope of this book, but is a prerequisite for some of the web-based groupware applications described in this chapter.

Complex groupware often requires the deployment of a specialized server (such as a calendar server), and sometimes involves integration with a directory server (schema update), as is the case with eGroupware.

The focus in this chapter will be on groupware, calendaring, and collaboration solutions that meet the needs of companies like Acme Widgets and Ballstyx Engineering.

Understanding Exchange and Outlook Groupware and Calendaring Features

The groupware and scheduling capabilities of the Exchange / Outlook combination are quite appealing. Using Outlook as a client and Exchange as the communication / management framework, enterprise groupware features become available to an entire organization.

Even without Exchange, the Outlook client contains a number of groupware or groupware-like features. Table 8.1 lists these features along with a short description.

Table 8.1 Outlook Features

Feature	Description
Contacts	names, e-mail/postal addresses, phone numbers, affiliation, birthday, etc.
Calendar	daily/weekly/monthly schedule views, appointments, reminders
Tasks	status, assignee, start/end dates, % complete, billable hours
Journal	work type, start/end times, affiliation, description

The Outlook calendaring is full-featured, with support for recurring appointments with multiple attendees, and support for the iCal and vCal standards (RFC 2445). When Outlook is paired with an Exchange server, the calendaring becomes multi-user, and adds the features listed in Table 8.2.

Table 8.2 Outlook Features with Exchange Server

Feature	Description
Shared Calendars	Exchange enables sharing of user and group calendars. Exchange enforces access control list (ACL) security restrictions.
Meeting Scheduling	Outlook allows a user to schedule a meeting and invite multiple recipients. When paired with Exchange, Outlook also shows the free/busy times of each attendee, and can even auto-pick a time when all attendees are available.
Resource Accounts	Allows the scheduling of resources such as meeting rooms, catering services, and projectors or other equipment.
Automated Request Processing	Enables the automation of specific scheduling functions. For example, a resource account (such as an infrequently-used meeting room) may be configured to automatically accept meeting invitations from anyone.

As you'll see in the next section, most of these capabilities have open source equivalents.

Understanding Linux-Based Groupware and Calendaring Services

Although there is no current open source drop-in replacement for Exchange calendaring, there are a number of solutions that meet the groupware and calendaring requirements for nearly all businesses. If your needs are not met by these solutions, you may consider using proprietary software for these capabilities, although the standard disadvantages of proprietary software apply.

In the sections below we will explore eGroupware and Horde, two of the more popular, mature, and full-featured groupware / scheduling open source applications. We also briefly mention OPEN-XCHANGE, which became an open source application a few weeks prior to the publication date of this book.

Understanding eGroupware

There are numerous examples of open source groupware software, but one of the most popular is eGroupware. eGroupware is a web-based suite that runs on the LAMP architecture. eGroupware requires Apache, MySQL or PostgreSQL, and PHP.

The feature set of eGroupware is comprehensive, and includes groupware features such as calendaring, and some not-so-groupware features, such as comics! A partial list of the many great features of eGroupware follows:

- Calendar, featuring personal and group schedules, alarms, and notifications
- Webmail (requires IMAP – Internet Message Access Protocol – server)
- Address book
- Trouble ticket system
- WikiWikiWeb
- Intranet portal (called Site Mgr)
- Project management
- Internet bookmark management
- Comics

eGroupware is a mature solution, and reached the version 1.0 milestone in August 2004. It is widely deployed, and the modular framework offers the ability to easily customize the solution. eGroupware features an attractive interface, support for multiple languages, and an active development team.

More information about eGroupware is available at www.egroupware.org. A working demonstration is available at www.egroupware.org/demo.

Understanding Horde

The Horde Application Framework is a web-based groupware framework that runs on the LAMP architecture. Horde requires Apache, MySQL or PostgreSQL, and PHP / PEAR (PHP Extension and Application Repository). The Horde framework provides common ways of handling inter-process communication, user preferences, browser detection, access control, user help, and more.

There are a number of groupware and groupware-style applications that use the Horde framework. Table 8.3 lists selected Horde Project applications along with a short description.

Table 8.3 Horde Project Applications

Horde Application	Description
IMP	Webmail access to IMAP or POP3 (Post Office Protocol v3) mailboxes
MIMP	Mobile phone/PDA (personal digital assistant) mini-version of IMP
Ingo	E-mail filter manager for procmail / Sieve
Forwards	Configures e-mail forwarding
Vacation	E-mail auto-responder for vacation notices
Turba	Address book / contact management
Kronolith	Calendar viewer
Hermes	Time-tracking system
Whups	Web Horde User Problem Solver (ticket tracking)
Rakim	Web chat
Nag	Task list manager
Tream	Bookmarks manager
Chora	Web-based CVS (concurrent versions system) viewer
Gollem	Web-based file manager

These applications provide significant functionality under a single framework. While not all of the applications will be strictly required by organizations, many of these are nice-to-have applications that are easily installed / integrated with the rest of the framework.

More information about the Horde framework and Horde Project applications is available at www.horde.org. A demo of popular Horde Project applications is available at www.horde.org/demo/.

Configuring & Implementing

Customizing Groupware Solutions

Most open source groupware and calendaring solutions (including Horde and eGroupware) are made up of an application framework with separate modules that provide each groupware feature. An advantage of this approach is that modules may easily be created, modified, installed, and removed. This greatly simplifies the process of customizing an organization's groupware solution.

Reviewing OPEN-XCHANGE (OX)

Unfortunately, due to the fact that OPEN-XCHANGE was put under open source licensing terms just prior to the publication date of this book, there was not enough time to allow for research and testing of the new open source version of this groupware collaboration server. Therefore, a full analysis of this product is not included in this book. Instead, a brief review of OX follows.

OX is the open source version of Suse's commercial Openexchange server. OX supports multiple languages, and provides the following features:

- Web portal
- Calendar services
- Contact management
- Task management
- Project management
- Bookmark management

- Web forums
- Webmail

According to the documentation, OX works with any browser, iCal, e-mail, or WebDAV client. Because OX uses Java and Java servlets, compiling OX requires the Java SDK, a Java compiler, Apache ANT, and a number of common Perl and Java libraries.

Because of the newness of this open source solution, it is too early to determine how this product will pan out. If OX lives up to its feature set, it may become the best open source replacement for Exchange. Openexchange is a mature product, so OPEN-XCHANGE is also likely to be reliable.

More information about OX is available at www.open-xchange.org.

Understanding TWiki Collaboration

Up to this point we have been primarily discussing calendaring and scheduling solutions. There is another variety of groupware solutions called *control management* and *knowledge management systems*. These types of groupware allow for collaborative creation and administration of content, typically in a structured way, following some type of workflow. They may also facilitate the creation, dissemination and utilization of knowledge.

WikiWikiWebs, or Wikis for short, are a popular way to create and maintain content and knowledge collectively by a group of people. A Wiki is a writeable web, a collection of connected web pages that change and grow “organically”. In most cases, editing of any page is allowed (and encouraged), even if you’re not the owner of the page. Wiki uses an easy-to-learn syntax that is much simpler than HTML (Hypertext Markup Language), although some Wiki implementations also allow the use of HTML tags. Editing is usually text based, although some also offer WYSIWYG editing.

One of the best-known Wikis is Wikipedia at www.wikipedia.org. Wikipedia is a free encyclopedia, where thousands of active contributors work on over 1 million articles in many different languages. Anyone on the Internet is invited (anonymously if desired) to edit any article on Wikipedia, or contribute his/her own article. There appears to be a complete lack of security and it sounds like an invitation for trouble. Graffiti happens occasionally, but it gets removed usually within minutes because many eyeballs are watching the content. This “soft” security based on version controlled content and peer review works quite effectively, even at the scale of the Internet. For more information about why Wikis work so well, navigate to <http://c2.com/cgi/wiki/WhyWikiWorks>.

A particularly useful Wiki “flavor” in corporations is TWiki, authored by Peter Thoeny. TWiki has more of a business collaboration slant than many of the other Wiki flavors, making it especially appropriate for the kind of companies represented in this book. For one, it can be used like any other Wiki to maintain content in a free and unstructured form. TWiki has features needed in this environment, such as authentication, access control, complete audit trail, document management, e-mail (and other enterprise systems) integration, and application plug-ins. This allows teams to maintain content in a structured way. TWiki excels at meeting common business requirements such as the following:

- A shared notebook for projects: repository, scheduling, meetings
- A departmental collaboration tool: process documentation, project reviews, QA tracking
- An Intranet publishing tool: IT, HR, company standards, policies and procedures
- A content management system with a focus on free-form collaboration: requirements capture
- A knowledge base: problem/solution pairs with attached patches
- A book-authoring tool: Author, review, and edit chapters of a book. This book has been authored collaboratively using TWiki
- A platform to create web based applications such as employee news portals, inventory systems, and issues tracking systems

Organizations are using TWiki for many different types of purposes, as demonstrated by the success stories at www.twiki.org submitted by Disney, Motorola, SAP, Yahoo, and others. For more information on Wikis and TWiki, consult www.wikipedia.org/wiki/Wiki and Twiki.org, respectively.

Migrating to Linux Groupware / Calendaring Services

Now that you have successfully tested and deployed groupware services, you may find value in migrating information from Exchange to your new groupware / calendaring system. Depending on the import / export capabilities of your groupware solution, this may or may not be feasible.

If the number of users is small, often the easiest method is to perform the migration manually, especially if the usage of calendar/groupware services is light. Although this “low-tech” approach is not characteristic of system administrator behavior, sometimes the simplest solution is the best solution. Many groupware and calendaring migrations, even in companies of 100 or more people, can be quickly performed using manual data entry, especially if the group that uses these features is small.

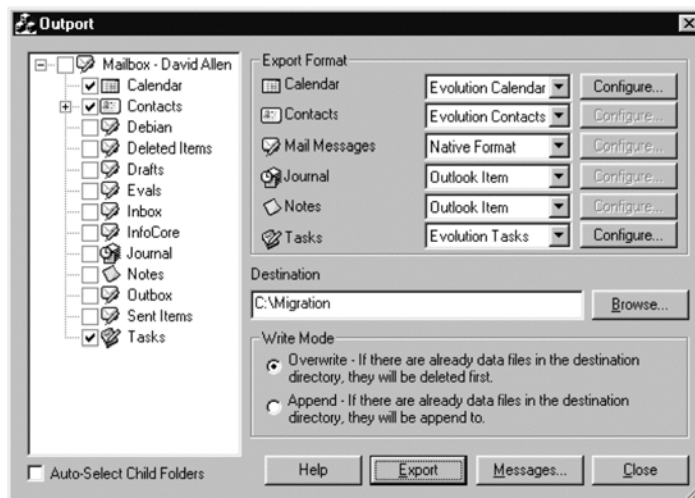
There are a number of standards, file types, and RFCs that cover some of the aspects of groupware, namely for contacts and calendar entries. These standards include iCal/vCal (RFC 2445) and vCard (RFC 2426). iCal and vCal are file representations of calendar entries, and a vCard file is a representation of a business card. iCal files generally use an .ics file extension, while vCal files generally use a .vcs file extension. vCard files have a .vcf extension.

Understanding Outport

Outport is one of the most useful applications for migrating contact, calendar, and task information from Outlook. The data can be exported the file formats of Evolution or other calendar/groupware clients. The word *Outport* is a concatenation of *Outlook Export*, which precisely describes what this program does.

Outport is very simple and easy to use. The left side of the window contains the folder tree for Outlook, with checkboxes to select each folder. The right side of the window controls the export configuration. Figure 8.1 shows a screenshot of Outport ready to migrate a mailbox.

Figure 8.1 Exporting Data Using Outport



This configuration will export the contacts, calendar, and tasks for David Allen's mailbox into a format understood by Evolution. The files are copied to the directory where Evolution stores its data, and the migration is complete.

Understanding Other Migration Methodologies

If you are migrating away from an e-mail/PIM (Personal Information Manager) client other than Outlook, Outport will not work for you. All is not lost – you still have a few choices if you don't want to perform your migration by manually entering data.

One of the most effective approaches for migrating calendaring, contact, and other groupware data is using an intermediary file format that is understood by both Windows and Linux applications. For calendaring, using the iCal/vCal (.ics/.vcs) file format works with almost all calendaring clients. Most clients that manage contact information can use files in the vCard (.vcf) format.

For objects that have no standard file type for information interchange, a text file representation can often be used to migrate the data from one application to another. If the applications support it, data may be output to a .csv (comma-separated values) or tab-delimited .txt file. This type of file is preformatted into rows and columns, and may be easily imported into a database-like structure. If there are a large number of migrations, it may make sense to write a Perl (or Python) script to parse the data from the original export and create an import file with the data properly reformatted for the target application.

While setting up file-based import/export requires an up-front time investment, it can be a useful approach, particularly if the problem set is large and homogenous. In other cases manual entry may be used, especially for a small workload. However, manual entry introduces the potential for human error ("fat finger" typing) and may not be a cost-effective use of a system administrator's time.

Summary

There are a myriad of groupware applications available for Linux, primarily running on a LAMP platform. These solutions duplicate most of the calendaring and groupware features of the Exchange-Outlook duo, although not always in the same integrated fashion as Exchange.

eGroupware and Horde are two very popular groupware suites. Both of these groupware suites provide a wide variety of features and capabilities, with each suite having its own strengths and weaknesses. OPEN-XCHANGE, the open source version of Openexchange, was placed under an open source license very close to the publication data of this book. OPEN-XCHANGE looks promising, but has not been fully reviewed.

When migrating from Outlook / Exchange to open source messaging servers and clients like Evolution, Outport (<http://outport.sourceforge.net>) can be used to easily perform the migration. Outport supports many types of file formats, and can be used to export data to almost any e-mail client / PIM.

Another migration method involves manual entry of data into the new calendaring / groupware system. This method has the advantage of being a potential option in all cases, but is only viable if the workload is small. It also introduces the potential of tying errors, and may not be a cost-effective use of a system administrator's time.

With the migration of calendaring and groupware complete, the final Exchange dependencies in an organization are removed, and the Exchange environment (and dependencies) may be retired.

Solutions Fast Track

Understanding Exchange and Outlook Groupware and Calendaring Features

- ☑ Outlook provides a number of groupware features, including a calendar, contacts, task list, and a journal.
- ☑ When Outlook is paired with an Exchange server, the combination delivers shared calendars, meeting scheduling, resource accounts, and automated request processing.

- ☑ Most features of Outlook / Exchange can be provided by open source software.

Understanding Linux-Based Groupware and Calendaring Services

- ☑ eGroupware is a PHP groupware and calendaring solution. eGroupware runs on a LAMP architecture, and integrates with OpenLDAP. More information is available at www.egroupware.org.
- ☑ Horde is a PHP/PEAR groupware application framework. The Horde Project contains a number of applications that provide groupware and groupware-like features. More information about Horde is available at www.horde.org.
- ☑ OPEN-EXCHANGE (OX) is an open source derivative of Suse's Openexchange server. Visit www.open-xchange.org for more information about OX.
- ☑ TWiki is a Wiki collaboration system that can be used for knowledge management and content management. More information about TWiki is available at www.twiki.org.

Migrating from Exchange and Other Groupware / Calendaring Services

- ☑ If you are using Outlook, the best way to migrate calendar, contact, journal, task, and note information is using Outport, available at <http://outport.sf.net>.
- ☑ If you are using a PIM client other than Outlook, you may use .ics, .vcs, and .vcf files to accomplish calendar and contact data migration if the source and target application support these formats. In other cases, text files or manual entry may be used.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- Q:** I can't find an open source application that has a particular groupware feature that I want. What do I do?
- A:** In some cases a proprietary solution (such as Bynari or Skalix) may provide the desired features.
- Q:** I want to run a groupware server that integrates with Outlook seamlessly. What do I do?
- A:** Consider using a groupware solution that features a plug-in for Outlook and/or Evolution.
- Q:** Where can I find more information about how to set up a LAMP architecture?
- A:** “Building a LAMP Server” at www.brtnet.org/linux/lamp.htm contains step-by-step instructions.

Web Services: IIS vs Apache

Solutions in this Chapter:

- Background: HyperText Transfer Protocol
- Understanding Microsoft's Internet Information Server
- Understanding Apache Web Server
- Migrating Static Sites from IIS to Apache

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Since the early days of the Internet, no single protocol has taken off quite like the HyperText Transfer Protocol (HTTP). In August of 1991, Tim Berners-Lee announced the first Web server that hosted files from The European Laboratory for Particle Physics (CERN). A year later, there were 50 web servers online. By 1999, there were over 720,000 public Web servers. In August 2004, according to the most recent Netcraft survey, over 54 million Web servers are accepting requests and this number is growing by 1 million every month. Of these, almost 70% of them are running Apache. The simplicity of the HyperText Markup Language (HTML) combined with ease of administration of an HTTP server allows anyone to host content for public viewing on the World Wide Web (WWW), perhaps some day everybody will be running one.

People deploy Web servers in many different situations and for many reasons. Many corporate businesses, independent consultants, as well as do-it-yourself end users have established lucrative businesses providing the technical know-how. You can find the fruits of their labor running on big installation servers, small network appliance firmware, and even the most unlikely places, like a sump pump. With some companies netting millions of dollars for their web applications and others simply upgrading their current systems to be web-“aware”, it’s no wonder that the desire to run a Web server is found in any technologically minded business plan. And the cost of running a Web server is relatively small compared to other means of advertising, such as ads in magazines, newspapers, brochures, or local yellow pages; people can easily publish information about their company’s business and products without much prior knowledge about the Web at a percentage of the cost of advertising in print mediums. Virtually any company that looks into web-enabling their business has something to gain. From the business standpoint, it seems to be a smart investment.

To the technologically minded, geeky, self-motivated individual, a Web server and robust operating system (OS) is a chemistry set of potential. Server side scripting languages, shell programs, and standard UNIX- based utilities can turn a Web server into a weather reporting station, a community bulletin board system, or a dating service.

This chapter is about migrating Web services from Windows Internet Information Server (IIS) to Apache Web Server and will provide an overview of Web services on Linux. Server-oriented Linux distributions package a plethora of utilities with Apache – we’re assuming it’s already clear to you what is gained by

migrating: a freely available operating system, accompanying services, limitless documentation, how-tos and FAQs, and broad user adoption, to name a few perks. By the end of this chapter you will see that virtually anyone can learn to administer his or her own Web server.

We will not delve too deeply into the programming side of web hosting, but we will introduce places where you can continue your research. This chapter will focus mainly on the administration and hosting of static content as an introduction to get the administrator up to speed running Apache. No doubt many of you will have skipped ahead to this chapter, so let's get to it, but first a little more history.

Background: HyperText Transfer Protocol

Back in 1989 Berners-Lee wrote a proposal for CERN for managing their information resources. The proposal incorporated earlier Gopher ideas (of a navigable tree of linked items), Ted Nelson's idea of "hypertext" from his work with Xanadu, and added three new important ingredients: a protocol for transmitting documents over TCP/IP (Transmission Control Protocol/Internet Protocol) connections, a markup language named HTML (HyperText Markup Language), and client software for viewing content written in HTML. The idea was to eliminate redundant information throughout CERN by "hyper linking" to other relevant documents to prevent repeating the information they contained.

Circularly cross-referenced material made a non-linear research path. A researcher could follow a hyperlink to another text, then another, then another, and eventually come right back to where they started. This feature enabled members at CERN to determine that the information that they were presenting was not already archived somewhere else.

Shortly thereafter, in 1993, Marc Andreessen led a team, working on a project for the National Center for Supercomputing Applications (NCSA), that made the first graphical World Wide Web browser. This web browser was called Mosaic. They then created their own Web server, the NCSA HTTP daemon (httpd). Later, Andreessen left the company and founded Netscape Communications Corporation with Jim Clark in 1994.

The idea of navigating through hundreds of documents by clicking a mouse was appealing and quickly took root. Netscape came out with its own server software, Netscape Communications Server, and a new market began.

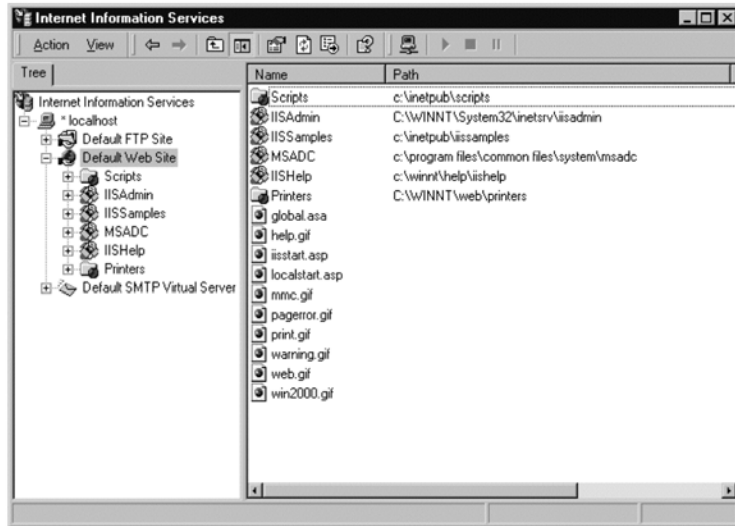
Naturally, Microsoft could not ignore this emerging technology now known as the Web, and developed its own web browser called Internet Explorer. Since HTTP is an open standard and no one held copyrights on the protocol, Microsoft started packaging its own version of a Web server called IIS.

Understanding Microsoft's Internet Information Server

Microsoft's Internet Information Server combines a few Internet services into one package. Along with the Web server, IIS also incorporates an outgoing e-mail, or SMTP (Simple Mail Transfer Protocol), server, and a file, or FTP (File Transmission Protocol), server. This section will provide an overview and some terminology for running web services with IIS 5.0. We will be using Windows 2000 Server. We will cover virtual directories, security and authentication, and we'll set up static web pages with IIS before we migrate our sites to Apache and Linux. You should know before you start that Internet Information Server is available in all of Microsoft's NT, 2000, 2003, and XP products, but some functionality is disabled in non-server flavors of these operating systems. Specifically, the ability to create multiple websites and the Digest Authentication method are disabled in non-server versions of Windows. SSL (Secure Sockets Layer) support is available in all, but cannot be configured without pre-purchasing a signature for your certificate request from a third-party Certificate Authority (please see the Apache section for creating self-signed certificates for use in Apache).

Getting Started

Administration of IIS Web Server is done through the **Internet Information Services** dialog box. Click **Control Panel | Administrative Tasks | Internet Services Manager** to access this dialog box. This allows you to configure the default FTP, Web, and SMTP servers, as shown in Figure 9.1. Once installed, via **Add/Remove Programs**, IIS is already running and listening for requests on port 80. Provided you don't have a firewall blocking port 80, you should be able to fire up a web browser and view the "Welcome to Windows 2000 Internet Services" page at <http://localhost>. This page, when viewed from the local computer, will prompt you with a Windows-based authentication login. Type in your user account and password and your browser will be redirected to a Getting Started page. If viewing from another computer, this page will simply display an "Under Construction" page to your users.

Figure 9.1 Internet Information Services Dialog Box

Default Index Page

To remove the “Under Construction” page and give your users something a little more interesting to view, simply create an HTML file called `Default.htm` and move it to the Web server’s home directory, which in IIS is `c:\inetpub\www-root`. You can use any text editor or WYSIWYG editor that you like to create `Default.htm`. After saving your `Default.htm` file, copy it to the Web server’s home directory and reload your web browser.

```
<!-- sample Default.htm page -->
<html>
  <body>
    <p>Hello world.</p>
  </body>
</html>
```

[warning]

Make sure that it’s called `Default.htm` and not `Default.htm.txt` because of the text editor you used. Turning file name extensions on in the operating system preferences will help safeguard against this doubled up filename extension. Go to **Control Panel | Folder Options | View Tab** and disable the **Hide file extensions for known file types** option to change this setting.

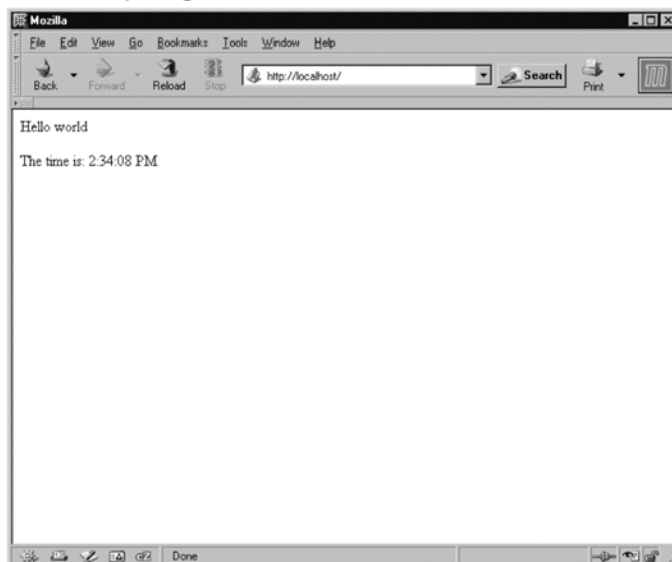
Other Web servers use a different convention for a directory's default index file, as we'll see with Apache. Usually this file is called `index.html`. IIS also uses `Default.asp` as a directory index. The filename extension `.asp` is an IIS convention that stands for active server pages (ASP). Active server pages are Microsoft's version of a server side scripting language. Alternatively, you can use `Default.asp` as a default index page for the Web server.

ASP is an embedded scripting language; regular HTML is processed just like regular HTML. To differentiate ASP commands to the Web server you need to surround your ASP syntax with `<% %>` characters. We're not going to cover ASP, but here's a quick example. For more information on scripting with ASP visit one of the numerous ASP portals on the Web such as www.asp.net.

```
<!-- sample Default.asp page -->
<html>
  <body>
    <p>Hello world.</p>
    <p>The time is: <%response.write(time())%></p>
  </body>
</html>
```

Now, copy `Default.asp` to `c:\Inetpub\wwwroot` and move `Default.htm` out of the way. Figure 9.2 illustrates how this will appear in the web browser.

Figure 9.2 Default.asp Page Viewed in a Web Browser

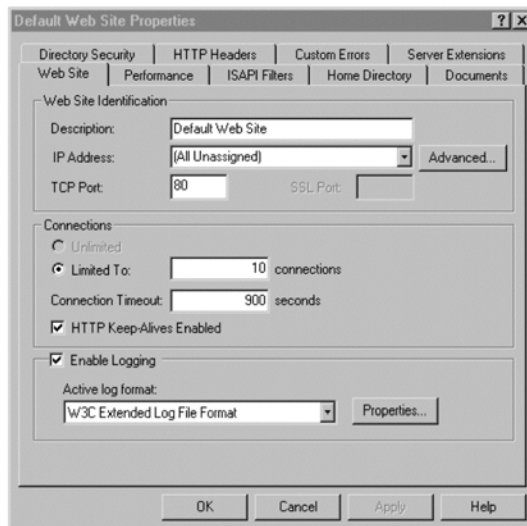


Default Web Site

Going back to the Internet Information Services dialog box and clicking on the **Default Web Site** part of the tree in the left pane, you will see that your Default.asp page now appears in the list on the right. By right-clicking on the right pane, or right-clicking on the **Default Web Site** branch on the left, you can access the **Properties** for the default Web server (see Figure 9.3). This is a context-sensitive dialog box, different for each virtual directory and virtual server. This dialog box allows you to define things like the home directory, default index page(s), and directory security settings and permissions. Take a look at these properties now and see what is available.

Under the **Home Directory** tab you will see the home directory listed as c:\inetpub\wwwroot – where we’ve been copying our default pages. Under the **Documents** tab you can list the default document to be displayed in this directory. Add index.html to this list. The **Web Site** tab shows that this default web site will be displayed for anyone accessing this server regardless of the canonical name they use to get there. Acceptable canonical names would be http://localhost, the computer’s name in the HOSTS file, the IP address, or the Fully Qualified Domain Name (FQDN). The **(All Unassigned)** field for the IP Address of this web site means that this will be a catch-all for HTTP requests coming in to this computer as long as they are not superseded by another catch-all or virtual server.

Figure 9.3 Default Web Site Properties



Virtual Directories

Virtual directories in IIS are real directories on a file system, but are special. In Apache, these are called *aliases*. They are used to map a directory portion of a URL (Uniform Resource Locator) to a special directory on the server's file system, outside of the Web server's home directory. The Web server knows where to look for files associated with a particular virtual directory by looking at the end of the requested URL (minus the filename). If the directory supplied is not a subdirectory of c:\Inetpub\wwwroot, then the Web server will check to see if it matches one of the virtual directory mappings. If it does match, the file requested will be searched for there.

Virtual directories are useful to separate information from the Web server's home directory. They can also be used if the directory you want to share is on another file system or another machine. Use the following steps to create a virtual directory to hold scripts outside of the Web server's home directory:

1. Create a c:\Inetpub\cgi-bin directory.
2. In the Internet Information Services dialog box, right-click **Default Web Site**, and select **New | Virtual Directory**.
3. A wizard will pop up asking you for the alias for that directory (the directory portion of the URL) and the real directory on the file system (the one you just created by hand). You will also be asked to set the access permissions in the wizard. The only permission that might not have an obvious meaning is the **Browse** permission. If you enable this option, the Web server will generate a directory listing if there is no default index page found. If disabled, your users will get a Directory Listing Denied page. We chose **Read**, **Script**, and **Execute** permissions since we know we are creating a script directory, and we don't want people to browse it. Now, we're all set up to use PERL CGIs or Server Side Includes (SSI) outside of our website's home directory – just drop your CGIs into the c:\Inetpub\cgi-bin directory and you can access them through a URL like `http://example.com/cgi-bin/Default.pl`, or `http://example.com/cgi-bin/Default.cgi`. Of, course, you will want to add these filenames to the default index files list if you are not using absolute URLs.

Security

Security in IIS is configured for the entire Web server, per virtual server, per directory, or per file. For each you can have *anonymous access*, *basic authentication*, *digest authentication* (or basic authentication with hashing), or *Windows-based authentication*. None of these are secure by themselves, but they get stronger when you layer them with SSL. The following illustrates the differences between them:

- **Anonymous Access** means there is public access to that resource.
- **Basic Authentication** an HTTP Internet standard authentication method, which is supported in many web browsers. It secures resources based on *realms* (per server, per directory, per file, etc...). Basic authentication uses a challenge-response process to establish that a user's credentials are correct and he or she is therefore allowed access to the resource. It provides only limited security, however, since passwords are transferred in clear text across the network.
- **Digest Authentication** meant to supersede basic authentication, but isn't much more secure. With it, the server sends a checksum to the to the browsing client. The browser then uses this and md5sum to develop a hash of the username, password, and URI (Uniform Resource Identifier) and sends them back to the server. The server can then use this information to reverse the process and check the credentials. This is a very simplified view of the process (see RFC2616 and RFC2617 for more information). Since the process is well known and reversible, this is only considered slightly more secure than basic authentication. Digest authentication is an HTTP Internet standard and has support in most web browsers.
- **Windows-based Authentication** is very similar to digest authentication, but it is a Windows standard and mainly supported in Internet Explorer. Some newer versions of Netscape and Mozilla also support Windows-based Authentication.

Generally, these authentication methods should only be used on internal networks since they do not provide strong security. You should never use these authentication methods on untrusted networks without SSL encryption.

To setup basic authentication for the default Web server:

1. Access the Internet Information Services dialog box.
2. Right-click **Default Web Server**.
3. Select **Properties**, then select the **Directory Security** tab.
4. Under the **Anonymous access and authentication control**, click **Edit**.
5. Deselect **Anonymous access**, and enable **Basic authentication**.
6. Select **Edit** and type in the domain name or machine name for the server.
7. Restart your Web server and reload your web browser. You will be prompted for a username and password. The username and password correspond directly to actual users on the system and are configurable via **Control Panel | Users and Passwords**.

To enable digest authentication, simply enable the digest authentication checkbox. No other configuration is necessary.

Enabling Windows authentication is almost identical to the former two methods. It uses a *cryptographic exchange* with the Internet Explorer browser and so should only be considered when you are working with a known, homogeneous network of Windows computers or fairly recent browsers that support this authentication method. To enable Windows authentication, simply enable the Windows authentication checkbox. Users are configured in the **Control Panel** under **Users and Passwords**.

One big limitation of these authentication methods is that there is no distinction between system user and HTTP user; using any of these as an authentication method for thousands of users on a web portal simply will not scale. These authentication methods are intended to be quick and simple, like a latch on a bathroom door. Reserve using these authentication methods to trusted intranets or home networks. To create more robust authentication methods, look into implementing a script-based (PERL, PHP, ASP) methodology that accesses a database for user information. These are very scalable and easily coupled with SSL.

SSL/TLS

Secure Sockets Layer and *Transport Layer Security* (TLS) are the Internet standards for encrypted HTTP (or HTTPS) communications. Although these are two separate aspects of encryption over HTTP, they are used hand-in-hand and most people simply refer to encrypted HTTP as “SSL”. Netscape Communications

Corporation designed the first Secure Sockets Layer protocol in 1996, and the Internet Engineering Task Force complemented it with TLS. For more information about TLS see the Internet Engineering Task Force's website at www.ietf.org (specifically, RFC2817 and RFC2818), and for more information on SSL, see the Internet Draft on "The SSL Protocol" at <http://www.netscape.com/eng/ssl3/draft302.txt>.

SSL support in non-server flavors of IIS cannot be configured unless you have purchased a signature from a Certificate Signing Authority (CSA). You can, however, create certificate signing requests (CSRs) with IIS. If you wish to run IIS with SSL, you can create this CSR in a wizard (right click **Default Web Browser**, select **Properties | Directory Security** tab, then **Server Certificates**). Then copy the resulting text file to a web form at your chosen certificate authority's website. They will provide you with more information about turning on SSL support in IIS after you purchase a signed certificate.

If you are using a server variant of Windows, you can use the certificate server to retrieve CA certificates, generate new certificate signing requests, or check on a pending certificate. This is the recommended way for handling your IIS server certificates, however IIS is not able to handle self-signed certificates for testing SSL connections with your server. To access certificate server, select **Add/Remove Programs** in the Control Panel, click **Add/Remove Windows Components**, and make sure that **Certificate Services** is installed on your computer. Then, open a web browser and access <http://localhost/certsrv>. You will be presented with an ASP-driven web wizard that walks you through the process of handling your certificate requests, as shown in Figure 9.4.

Figure 9.4 Microsoft Certificate Services Welcome Window

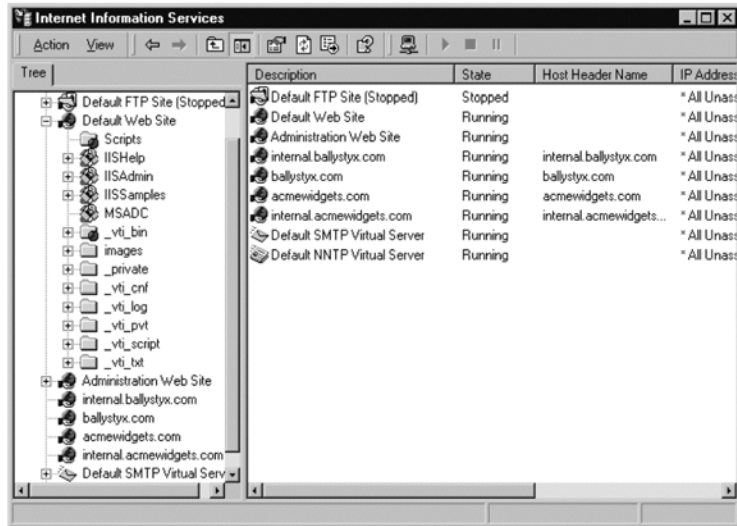


Virtual Servers

If you've made it this far with web services using NT Workstation, 2000 Professional, XP, or XP Pro and you now want to create another website (or virtual server), we're sorry, you can't – well, you're not supposed to. The ability to create a second website is a feature disabled in those products and is only available in the server flavors of Windows NT/2000/2003.

If you are using a server variant of Windows OS, new sites are created through another wizard in the **Internet Information Services** dialog box. However, before you add a new site, first decide if you are setting up a name-based or an IP-based virtual server. Name-based is when you have several domains pointing to one IP address and IP-based is when you have several IP addresses pointing to one computer. Name-based virtual servers don't need any special consideration, but if you are doing IP-based, then you will need to configure the network interface(s) for each IP that computer controls. To configure the interface(s) for multiple IP addresses access the **TCP/IP Properties** for the interface(s) in your computer and click **Advanced** to add more IPs. What follows assumes you have at least one public IP address through your ISP (Internet Service Provider) and you have already configured DNS (Domain Name Service) or edited the HOSTS file to point to that IP for each domain you wish to serve.

Configuring IIS to recognize the domain and redirect requests to the proper directory is done in the **Internet Information Services** dialog box. Right-click on the main server branch, then select **New | Web Site**. You'll be presented with a wizard and asked to type in the **Name** for the site which is any identifying name, (just for organizational use in IIS), the IP address of the computer, the port number you want that virtual server to run on, and the Host Header name, which is the domain name you are serving. On our system we created two domains, `internal.ballystyx.com` and `www.ballystyx.com`, and set the **Name** and **Host Header** the same for both. We then associated the domain with a directory on the file system. Like virtual directories, this directory, seen in Figure 9.5, can be anywhere; we created the directory `c:\inetpub\vhosts` on our system to put our domain directories.

Figure 9.5 Internet Information Systems Directory

Now, create web pages and copy default index files to your new virtual server directories. Once done, test to make sure you didn't miss a step. We made two different pages, one for `internal.ballystyx.com` for internal use and one for `www.ballystyx.com` for the public, and then copied them to `c:\Inetpub\vhosts\internal.ballystyx.com` and `c:\Inetpub\vhosts\www.ballystyx.com`, respectively. You can see what these pages look like in Figures 9.6 and 9.7.

Figure 9.6 Internal Default Index Files for `internal.Ballystyx.com`

```
<!-- Example Default.htm for internal.ballystyx.com -->
<html><head><title>----- Ballystyx Internal -----</title></head>
<body bgcolor=#FFFFFF>
<center>
    <img src=ballystyx_rocket.gif border=0 alt=logo>
    <h1>Greetings Programs</h1>
    <p>Here you will find resources to make your job eaiser.</p>
    <br>
    <br>
</center>
</body>
</html>
```

Figure 9.7 Public Default Index Files for www.ballystyx.com

```
<!-- Example Default.htm for www.ballystyx.com -->
<html><head><title>Ballystyx Global Semiconductor Engineering</title></head>
<body bgcolor=#FFFFFF>
<center>
    <img src=ballystyx_rocket.gif border=0 alt=logo>
    <h1>Welcome To Ballystyx Semiconductor Engineering</h1>
    <p>We welcome you to our new site on the web.</p>
    <br>
    <br>
</center>
</body>
</html>
```

NOTE

- **Index Page** This is the default page the Web server pulls up if an absolute page is not provided in the URL.
 - **Virtual Server** One of potentially many hosted domains on one Web server.
 - **Name-based Virtual Server** Many domains resolving to one IP address.
 - **IP-based Virtual Server** Many IP addresses resolving to one computer.
 - **Virtual Directory** A URL alias to a directory. It maps a special path that you want in a URL to a directory on the filesystem.
 - **SSL/TLS** Internet standard for encrypted HTTP.
-

Understanding Apache Web Server

Configuration of Apache, by comparison, is much more transparent. You can install and run it on Windows and UNIX/Linux as well as a host of other operating systems, including embedded systems. It supports multiple virtual servers, called *virtual hosts* in Apache-speak, and important configuration options like authentication and SSL can be set up easily. Configuration is traditionally done

through text files. However, we'll mention some useful graphical administration tools as well; both local software, packaged with a distribution, and web-based administration tools such as WebMin, are available to the system administrator.

The Linux community has a history of distribution wars. Since the source code is available to thousands of software applications licensed under the GPL, many competing distributions have evolved over the years. Distribution wars aside, we chose to use Fedora Core (<http://fedora.redhat.com>) for this section.

The Fedora Project is a RedHat-sponsored and community-supported open source project. It is also a proving ground for new technology that may eventually make its way into Red Hat products. It is not a supported product of Red Hat, Inc. Even though Fedora is not officially supported, they have made available many features to end users, empowering them to *get involved* with the distribution. This is a lot like Debian project (<http://debian.org>); the results of the efforts are in the public domain, but the distribution itself is based on the RedHat Package Manager (rpm) instead of Debian Packages (dpkg).

Background of Apache

Apache Web Server is the most ubiquitous server on the Web and it shows no sign of disappearing. According to recent NetCraft statistics, Apache is in use by nearly 70% of the total Web servers on the Internet. IIS is second with a little over 21%, and all other Web servers, including Sun, iPlanet, and httpd, make up the final 10%. These are exciting times for Apache administrators. Coupling this with the fact that Apache is currently developed and maintained by the Apache Software Foundation as an open source, community-based project, in the public interest, you have an excellent real-world example of how an open source solution can surpass proprietary development models.

Apache grew out of the NCSA's httpd software in the form of patches to the code. In February 1995, the NCSA's httpd was the most popular Web server on the Internet, but when the main developer, Rob McCool left the company, development lapsed. A small group of developers, including Brian Behlendorf and Cliff Skolnick, gathered these patches to the original httpd code and released the first version of Apache, version 0.6.2, in April of 1995. The name is a play on words: the resulting software was considered to be "a-patchy" Web server. Since then, the code has been completely rewritten and there are now eight core developers and a host of contributing authors who formed the Apache Group, which later became the Apache Software Foundation.

Installation

Installing Apache on Linux systems is very easy. Depending on which distribution you are working with, it might already be installed. If it is not, check to see if there is a pre-made package available for your particular distribution. In this section we'll be installing and configuring the most recent major release of Apache, version 2.0.

To see if Apache is installed on Debian, access a command prompt and enter the command **dpkg -l | grep apache**. On Fedora and RedHat-like systems, use **rpm -qa | grep apache** or **rpm -qa | grep httpd** (some systems name the package “httpd” instead of “apache”). On other systems, locate `apache`, and if a large list of files scrolls across the screen, you can be pretty sure that it is installed. See the documentation for your particular package manager for more information on listing packages and determining installed packages.

If you don't have Apache installed, locate the appropriate package for your distribution and install it. You can find RedHat rpms at <http://rpmfind.net> and <http://freshrpms.net>, and Debian packages are available through Debian repositories, which are usually pre-configured into the OS and accessible through the *apt* utilities. If the utility *apt-get* is available on your system, you can issue the command **apt-get install apache** or **apt-get install httpd**, and *apt-get* will download a package and install it for you. Otherwise, go to the Apache Server's download page and download the source code (<http://httpd.apache.org/download.cgi>). This section will not cover compiling Apache from source code – there are many documents on the Internet that cover this topic. We recommend using pre-built packages whenever possible, since in most cases there is a package available for your Linux distribution already, and a plethora of complimentary packages that go along with it. Once you have a package for your system, run the package manager tool to install it. On Debian systems this would be something like **dpkg -i apache-2.0.x.dpkg** and on RedHat systems this would be something like **rpm -i httpd-2.0.x.rpm**. The actual package name will vary and even though many packages will work on different distributions, we recommend tracking down the package specifically built for your system.

Starting, Stopping, and Checking Status

Starting Apache is done through the system's rc style init scripts in `/etc/init.d`. To start Apache type the following command as root at the command line **/etc/init.d/httpd start** or **/etc/init.d/apache start**. To stop Apache type

`/etc/init.d/httpd stop` or `/etc/init.d/apache stop`. Whether this script is called `httpd` or `apache` depends on your distribution. The script will also accept other commands such as **status**, **reload**, **help**, and **configtest**. Run the script by itself, with no arguments, to see which commands are supported.

Apache also provides its own administration script, *apachectl*. It accepts commands such as **start**, **stop**, and **restart**, and others. Run `/usr/sbin/apachectl` by itself or check the manpage (manual pages) for more information.

Configuration

It would be way too slow and painful to go through every configuration option available to Apache (there are numerous), so we will just cover the basics. We'll explain the components of the main configuration file, *httpd.conf*, and how to configure the options we are now familiar with. We'll discuss including other configuration files as well as Apache modules: which modules come with Apache, and how to include more. Along the way we'll point out some graphical tools for easing the task of administration and mention other configuration options where they apply.

Configuration of the Apache server is done through the file named *httpd.conf*, but before we discuss how it works, let's take a look at the directory structure that file assumes. Depending on compile time options, this file can be located in different places. More often than not it is in the `/etc/httpd/conf` directory. For the purposes of this section, we are going to assume a RedHat-based installation that puts the `ServerRoot` in `/etc/httpd` and the `DocumentRoot` in `/var/www` – more on these terms in a minute. On a Fedora Core 2 system, after the `httpd` package's installation, the `/etc/httpd` directory contains five items. These items are two configuration directories: `conf` and `conf.d`; and three symbolic links pointing to other places on the server's filesystem: `logs`, `modules`, and `run`.

We'll explain these directories in turn: `/conf` contains *httpd.conf*, the main configuration file for Apache; *magic*, which is a file that helps Apache ascertain filetypes; several `/ssl.*` directories for certificates, keys, and requests; and a *Makefile* that can be used for generating certificate signing requests and self-signed certificates to use with the server. The `/conf.d` directory contains separate configuration files that can be separated for logical organization, but could just as easily be part of the main configuration file. The files are included by a wildcard *Include* in the *httpd.conf* file. Modules and other packages that require a special Apache configuration option will write their configuration files `conf.d`. The `/logs` directory contains server logs, `/modules` is the module directory, and `/run`

is a symlink to your distributions equivalent of a `/var/run` directory, which is where many programs write their process ID (PID).

As we have already mentioned, different Linux distributions will put these directories in different places, and if you install from source you have the option of choosing where these directories will be by supplying the `-f` option to the `./configure` line. See the source documentation for how to do this. Note that if the Apache you are working with was installed from source, failing to explicitly tell Apache where to install will cause it to be placed in `/usr/local/apache`.

Okay, now that you've gotten the layout, let's dive into configuration file syntax.

The `httpd.conf` file has two main sections: the *Main* section and *Virtual Host* section(s). The Main section is anything not within a **<VirtualHost>** container. We will use these terms from now on. The Main section engulfs any virtual host sections, if any, in a nested fashion. In this way a virtual host can override, if allowed, a Main configuration option, but if no configuration is provided within that VirtualHost container, then the settings for that option will default to the Main section. Either section can have any number of *directives* and *containers*, and the latter may be nested further.

- **Directives** can be one per line. You can use a `\` (backslash) to wrap a directive to another line. Directives are in the form **Directive Argument1 Argument2 [...]**. Arguments to directives are case-sensitive, but directives are not.
- **Containers** are like markup language in that you have an open and close tag defining the beginning and end of a container with an argument to a container appearing in the opening tag. They are divided into two different types: *Filesystem* and *Webspace* (the Apache documentation uses these terms to describe the difference between the most common types of containers you will find in the configuration file). *Filesystem* containers, such as **<Directory>** and **<Files>**, operate on the local file system and control permissions and options for a whole directory and subdirectories, or a particular file. If you have a **<Files>** container in the Main section of the configuration file, then those permissions will apply to any file with that filename, regardless of which directory it is in. However, if you nest a **<Files>** container inside a **<Directory>** container, then those permissions will apply to any file with that filename under that directory and below. On the other hand, a *Webspace* container, such as **<Location>**,

operates on requested URL, ascribing permissions or actions based off of the path in the URL. A location doesn't necessarily need to be a directory on the file system either.

Another kind of container is the **<VirtualHost>** container. It is used to delineate the Main server (or default web page), from the virtual hosts, (or virtual servers as mentioned in the previous section). A

<VirtualHost> container can contain any number of directives, and file and web space containers, but certain combinations do not make logical sense. Apache is very good about alerting the administrator when directives are in conflict. You can use the **-t** option to `apachectl` to check your configuration syntax as well.

NOTE

Many containers have regular expression equivalents, such as **<DirectoryMatch>** and **<FilesMatch>**, that allow you to match on regular expressions instead of string literals.

Let's run through a few examples to solidify these ideas, but first one more note about comments and whitespace. Within the `httpd.conf` file, a pound (`#`) sign at the beginning of a line is a comment, and cannot be put in the middle or end of a line. Whitespace is ignored, even at the beginning of a line, so it is okay to indent your directives for clarity and organization.

Basic Configuration Options for the Main Server

Now that we have the basics of the Apache configuration file syntax down, let's take a look at a minimal `httpd.conf` file and apply these ideas. The actual file is over 1000 lines, but we removed all the comments and stripped the file down to what you would need to turn the server on. For the most part, when dealing with this file, the rule-of-thumb applies that if you don't know what the option does, leave the default and it's probably okay. In the past, Apache forced an administrator to edit this file, not turning on if they didn't. Today, a default install of Apache will start on the localhost, listening on port 80, and will serve documents out of `/var/www/html`. We'll explain these options after the example.

```

### Main Server Section
#
ServerName www.acmewidgets.com
ServerRoot "/etc/httpd"
ServerAdmin root@acmewidgets.com
DocumentRoot "/var/www/html"
User apache
Group apache
Listen 80

DirectoryIndex index.htm index.html index.php Default.htm index.cgi

### Directory Container for whole file system
#
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

### Directory Container for DocumentRoot
#
<Directory "/var/www/html">
    Options Indexes
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

```

WARNING

If you were to use the above configuration file, your server would not work. This configuration file is meant to illustrate what options you should look at immediately to turn the Web server on. Leave all other configuration options you see in the file as-is.

- **ServerName** is the default name for this server. It is a fully qualified domain name and can be different from the machine name. This is how Apache will identify itself to visitors.
- **ServerRoot** as mentioned earlier, is where Apache thinks it is installed on the file system. Most packages for Linux install Apache in the `/etc/httpd` directory, so the configuration files and logs are in this directory.
- **ServerAdmin** is the administrator e-mail displayed on error pages and in environment variables for scripts.
- **DocumentRoot** is the equivalent of the default home directory in IIS-speak. This is where the server will serve documents from if no virtual host is specified.

In the above configuration the user and group that the Web server runs as is “apache”. The *listen* directive can be an IP:port pair or just a port number. If the IP is not defined, Apache listens on all configured interfaces for that port. Multiple *listen* directives can be in one configuration file. In Apache 2.0, at least one *listen* directive must be present or the server will not start. We’ll see a more elaborate example of the *listen* directive in the “Virtual Hosting” section.

DirectoryIndex specifies the default index page to pull up if an absolute page is not specified in the URL.

The rest of this file is containers specifying the permissions and behavior of the root file system and the *DocumentRoot*. These are the most common lines to have inside of containers. Options can be: *ExecCGI*, *FollowSymLinks*, *Includes*, or *Indexes*, to name a few (there are couple more, but they are more esoteric). If *Options* is omitted from the container than *All* is assumed. *ExecCGI* says this directory is a CGI script directory. *FollowSymLinks*, means the server will follow symbolic links in this directory. *Includes* allows Server Side Includes. *Indexes* will generate a listing of that directory if no default index page is found.

The *AllowOverride* directive tells Apache which of these options may be overridden later in the configuration file or in a *.htaccess* file. *Allow* and *Deny* (not shown) directives define the hosts that are allowed to access this directory or resource. These can be listed be hostname, IP, IP range, or even by environment variables captured from the browsing client. The *Order* directive defines what order the *Allow* and *Deny* directives are applied.

Now, fire up your Web server with the following command and test with your favorite web browser:

```
/etc/init.d/apache start
```

Default Web Page

Your web browser should be displaying a “Test Page” or an “It worked!” page. The process to change the default web page is the same as in IIS. Simply copy a default index page to the *DocumentRoot* directory. In our configuration file, we defined *DocumentRoot* to be */var/www/html*, so copy it there and rename the page to something named by *DirectoryIndex* directive. In the next example we’ll make an *index.html* page for *www.acmewidgets.com* and copy it to the *DocumentRoot* directory (see Figure 9.8).

Figure 9.8 *DocumentRoot* Directory for *www.acmewidgets.com*

```
<!-- Example index.html for www.acmewidgets.com -->
<html><head><title>Acme Widgets</title></head>
<body bgcolor=#FFFFFF>
<center>
  <img src=acme_anvil.gif border=0 alt=logo>
  <h1>Welcome to Acme Widgets on the Web</h1>
  <p>We welcome you to our new home on the Web. We just secured a new
  hosting provider and they use 100% pure open source!</p>
  <br>
  <br>
</center>
</body>
</html>
```



Aliases and ScriptAliases

Identical to IIS's virtual directories, Apache provides for a way to map a requested URL path to another part of the file system. Configuring these in Apache is done with the *Alias* directive. The most common alias to have in the Apache config is the */icons/* directory. This is more of a historical thing when system administrators would try and save from having multiple copies of the same images all over the file system. A system administrator would use the icons directory as a single repository for images and then regardless of the web space that is being served, HTML `` tags can be written like the following example and Apache would know where to find that image.

```
<img src=/icons/blank.gif>
```

Apache sets this Alias up as follows in the `httpd.conf` file.

```
Alias /icons/ "/var/www/icons/"
```

```
<Directory "/var/www/icons">
```

```
Options Indexes
AllowOverride None
Order allow,deny
Allow from all
```

```
</Directory>
```

The *Alias* line tells Apache that any requested URL with */icons/* in the path should be mapped to the directory `/var/www/icons` and the requested file should be looked for there. The *Options* line that lists *Indexes* says create a directory index. No options can be overridden by other configuration files and the default is to allow anyone access.

ScriptAlias, by comparison, is the same as *Alias*, but the *ScriptAlias* directive implies that the alias will be serving CGIs. The directory container for that directory need not contain an *ExecCGI* option. Take a look at the following example.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

<Directory "/var/www/cgi-bin">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Security and Permissions

Apache's mechanisms for controlling security on the server are just as granular as, if not more so than, IIS. Apache has the ability to allow and deny hosts, file permissions, and basic and digest authentication to control access.

As we've already seen, the *allow* and *deny* directives let you to explicitly control individual hosts, or even entire networks accessing the site. The format is **Allow host1 host2 [...]**. A host can be an exact IP address, a partial IP address, a network segment or CIDR address, an IPV6 address, or the value of variable set by a script in the Web browser's environment, such as:

```
## Allowing only localhost to access files named admin.html
#
<Files "admin.html">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from 127.0.0.1
</Files>

## Allowing the internal network and localhost to access files named
# admin.html
#
<Files "admin.html">
```

```
Options None
AllowOverride None
Order allow,deny
Allow from 192.168.0.0/24 127.0.0.1
</files>
```

The *Deny* directive accepts the same syntax but denies any host that matches. File system permissions apply to the *User* and *Group* the Web server runs as. In our case we defined this as **apache**. Note, this must be an actual user and group on the system. In most cases your distribution will have defined this for you. If you are unsure, leave it set to the default in the configuration file). With file permissions, if the apache user does not have read, write, or execute permission on a directory, for example, then it cannot list, write to, or traverse that directory branch.

Generally, file permissions for the *ServerRoot* are owned by the *root* user. *DocumentRoot* directories are generally owned by *apache* or the Web server user. When Apache is started it up, it launches one mother process that runs as root, then it launches up to 10 child processes (set in the configuration file) that run as the Web server user. Logs and other process information are handled by the mother process, and the children handle requests. Therefore, your permissions must be set according to the applications you are running. If all you are serving is static content from the *DocumentRoot* directory, not much special consideration is needed – default permissions of 755 /var/www/html and owner of root should suffice.

```
drwxr-xr-x  2 root      root 4096 Sep 10 14:36 html
```

However, if you want to run scripts that write to the file system, or if you have a CGI directory that contains files that literally execute, then you need to make sure Apache has write access to the directory, or that the execute bit is set for scripts in your *cgi-bin* directory, respectively.

Apache also has basic and digest authentication methods just like IIS. These can be defined within containers and can be used in the *Main* section of the config, in *directory*, in *files*, and in *VirtualHost* containers. To set up basic authentication you will need to add these lines to one of those areas.

```
AuthType Basic
AuthName "Acme Widgets Internal Area"
AuthUserFile /etc/httpd/passwd/acme_internal
Require valid-user
```


The *AuthType* can be either **Basic** or **Digest**. See the earlier section of this chapter for the differences between them. *AuthName* is the name of the area that is protected. This name will appear in the pop-up window prompting the user for login credentials, so it's useful to make it meaningful for your users to know where they are logging in. The *AuthUserFile* defines where the passwords are kept. HTTP users are not users on the system, unlike IIS, there can be as many *htpasswd* files around as you need. You create new users and change passwords with the **htpasswd** command. We'll discuss this next. The *Require* line specifies which users can access this area. If set to **valid-user**, any valid user in that password file will be allowed access. This can also be a space-separated list of usernames in the file.

Administration of many HTTP users is not a pretty job. This facility was not designed to be scalable, but it is useful for adding a couple administrator accounts to a particular area or file. To create the password file, make a directory somewhere on the system where you want to store *htpasswd* files. We created */etc/httpd/passwd/passwords* for this purpose. Then create file, as root, while adding a new user to it with the following command:

```
$ htpasswd -c /etc/httpd/passwd/passwords admin
```

The **-c** option creates a new file and the username **admin** will be the only entry in it. This command is interactive, as it is and will prompt you for the admin's password. The password will be stored in the specified file encrypted with the md5 algorithm by default. If you want to create another user simply issue the following command:

```
$ htpasswd /etc/httpd/passwd/passwords acme
```

This will add the user **acme**. If you want to specify the password non-interactively, you can add it to the end of the command:

```
$ htpasswd /etc/httpd/passwd/passwords acmeadmin new_password
```

You can have as many password files as you like and/or keep them in different places depending on your needs. If you are doing domain hosting, for example, you would probably want to create a password file for each individual host, locate it in each host's home directory, and make it only readable only to that user and the Web server user.

.htaccess Files

Apache has the ability to allow users and administrators of sites on the server to override configuration settings for particular directories. The advantage of this is that a web programmer can be allowed to configure certain server options for his or her site without giving access to `httpd.conf`. To set it up, the main configuration file, `httpd.conf`, must have the *AllowOverride* directive defined for the **<Directory>** and there must be an `.htaccess` file at that location. *AllowOverride* accepts a space-delineated list of grouped option names that you want to allow to be overridden. The grouped option names can be: *AuthConfig*, *FileInfo*, *Indexes*, *Limit*, *Options*, *All*, and *None*. The `.htaccess` file is a text file containing configuration options for that directory. It has to be readable by the Web server. Note the period at the beginning of the filename. *AllowOverride* only works in **<Directory>** containers, but the `.htaccess` does not have to have another directory container in it. It only has to place the options relevant to the directory it is in.

```
# httpd.conf
<Directory /var/www/html/internal/>
    AllowOverride Indexes AuthConfig
</Directory>
```

In the above example, we are allowing the web programmer that works on the internal website to override *Indexes* and *AuthConfig* directives. Allowing *Indexes* means that the web programmer can override options relating to directory indexing in the directory where the `.htaccess` file exists. And allowing *AuthConfig* means that users can use directives that apply to setting up Basic and Digest Authentication for that directory.

The next example illustrates what the related `.htaccess` file might look like.

```
# /var/www/html/internal/.htaccess
Options Indexes
AuthType Basic
AuthName "Internal Use Only"
AuthUserFile /etc/httpd/passwd/internal.passwords
Require valid-user
```

If the web programmer has access to a shell on the server, and can write to the `/etc/httpd/passwd` directory, then creating a new password file and filling it with users is done just like above. But if the web programmer does not have write access to that directory, then he might decide to put their password file in

his own directory or another place the system administrator has provided outside of the Web server directories. Since the *Indexes* option is specified, then directory indexes will be generated for any directory under the `/var/www/html/internal` directory that does not have an `index.html` page.

If you want to change the name of the `.htaccess` file you can use the *AccessFileName* directive in the `httpd.conf` file.

Virtual Hosting

Apache was the first Web server to have support for virtual hosting. This facility, like virtual servers in IIS, allows you to host multiple domains with the same Web server. Apache virtual hosting is set up through the `httpd.conf` file and can be configured for any combination of name-based, IP-based, or both kinds of hosts. Name-based virtual hosting is configured using the *NameVirtualHost* directive and *VirtualHosts* containers. IP-based needs the *Listen* directive, to tell Apache which IP addresses to listen to, and *VirtualHosts* containers. Let's run through a couple examples and see what the differences are.

To turn on name-based virtual hosting, in addition to the *NameVirtualHost* directive in the Main section of the `httpd.conf` file, you also need, at the very least, a *ServerName* directive inside of each *VirtualHost* container so Apache knows to handle requests for that domain name.

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName internal.acmewidgets.com
    DocumentRoot /var/www/vhosts/internal
</VirtualHost>

<VirtualHost *:80>
    ServerName billing.acmewidgets.com
    DocumentRoot /var/www/vhosts/billing_dept
</VirtualHost>
```

In this example, Apache will serve requests for `internal.acmewidgets.com` or `billing.acmewidgets.com` out of `/var/www/vhosts/internal` and `/var/www/vhosts/billing_dept` DocumentRoots, respectively. Requests can come in on any IP on the server and through port 80. Apache deals with the requests by matching the *ServerName* directive in the *VirtualHost* container to the requested

server name in the URL. This example assumes that DNS is configured to point to the IP address of the server for these subdomains of `acmewidgets.com`. You can have as many domains or subdomains as you like as long as DNS is configured accordingly. The `VirtualHost` container can contain just about anything the `Main` section of the configuration file can so it is good to set up things like directory permissions, authentication, log files, and server administrators, just like you would be setting up a `Main` section.

To configure IP-based virtual hosting you need to first set up multiple IP addresses on the computer. The simplest way to add another IP address on a Fedora or RedHat variant system is by copying the `/etc/sysconfig/network-scripts/ifcfg-eth0` file to `/etc/sysconfig/network-scripts/ifcfg-eth0:0` and change the `DEVICE` variable to match in the new file.

```
$ cd /etc/sysconfig/network-scripts/
$ cp ifcfg-eth0 ifcfg-eth0:0
$ cp ifcfg-eth0:0 ifcfg-eth0:1
$ vi ifcfg-eth0:0
```

```
# Sample ifcfg-eth0:0 file
DEVICE=eth0:0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.0.53
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
```

```
# Sample ifcfg-eth0:1 file
DEVICE=eth0:1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.0.54
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
```

You can add as many IP addresses to the `eth0` device that you want by incrementing the number after the colon for each IP address and assigning the new IP address to the `IPADDR` variable. In the above example we used a class C

(private) address space. You can also assign more IP addresses to other interfaces by repeating these steps for `eth1` or `eth2` up to `ethN` interfaces.

Now that your IP addresses are configured, you can set up IP-based virtual hosting in Apache by adding another *Listen* directive to the Main section of the `httpd.conf` file. As stated earlier, Apache will listen to all interfaces if no IP is specified, but explicitly defining it maximizes predictability and greater control of how the server functions. In the next example we'll set up two IP-based virtual hosts, one on the default port of 80 and one on 8080.

```
# Main section of httpd.conf
Listen 192.168.0.53:80
Listen 192.168.0.54:8080

<VirtualHost 192.168.0.53:80>
    ServerName sales.acmewidgets.com
    DocumentRoot /var/www/vhosts/sales_dept
</VirtualHost>

<VirtualHost 192.168.0.54:8080>
    ServerName internal2.acmewidgets.com
    DocumentRoot /var/www/vhosts/internal2
</VirtualHost>
```

Apache has support for SSL/TLS encryption via the SSL module, *mod_ssl*. The *mod_ssl* module comes as a separate package on most distributions. On Fedora Core, for example, the package is simply named *mod_ssl* and depends on the *openssl* package. Whatever your flavor of Linux, it's best to hunt down the package that was built for your distribution. If you're on Suse, check on the installation disks; Debian, check the repositories with *apt*; RedHat/Fedora, go to <http://rpmfind.net> or <http://freshrpms.net>. Mandrake, probably has it on the installation disks or on <http://rpmfind.net>.

After installation, the *mod_ssl* package will have installed a *mod_ssl* shared object library and several configuration files in the *ServerRoot* directory. The shared library contains the functions that Apache needs to perform SSL communications and the configuration files control how SSL support is handled by Apache. There is also a default server certificate and a *Makefile* for generating new certificates, but we'll use *openssl*. First however, let's test our SSL-enabled Web server by restarting Apache and attempting to connect to <https://localhost>.

You should get a warning message (Figure 9.9) from your browser saying that the browser was unable to verify the identity. If you click **Examine Certificate** you can see the values for the default server certificate, as shown in Figure 9.10

Figure 9.9 Unknown Authority Warning Message and Options



Figure 9.10 Default Server Certificate Values



If you can see the warning, your SSL-enabled Web server is functioning. The default certificate, installed by Apache, is issued to "SomeOrganization". To create

a new certificate request, enter the following commands. You must have the openssl package installed to do this.

```
$ # backup your original private key and certificate
$ cd /etc/httpd/conf
$ mkdir ssl_bak
$ cp ssl.key/server.key ssl_bak/
$ cp ssl.crt/server.crt ssl_bak/
$
$ # make new Certificate Signing Request and private key
$ openssl req -new > acmewidgets.csr
$ openssl rsa -in privkey.pem -out acmewidgets.key
$
$ # alternatively sign the CSR yourself
$ openssl x509 -in acmewidgets.csr -out acmewidgets.crt -req -signkey \
acmewidgets.key -days 999
```

This creates your new CSR and the private key. Alternatively, you can sign the certificate yourself and use it to encrypt HTTP communications between your server and your client's browsers. Be aware, however, that this is not an official certificate. It is a "self-signed" certificate that was validated by you. If you, your intranet, or a small group of users are the only ones that will see this certificate, it is possible to tell your users to accept the warnings and proceed with their log in, or whatever you are redirecting to HTTPS for. On the other hand, if you are establishing an e-commerce store front and need encrypted HTTP to accept credit cards online, then you will want to purchase a real third-party signature from one of the many CAs on the Internet. These CAs are registered with popular web browsers and their CA certificates are distributed with the web browser package. This way, if your certificate is signed by a CA that your browser recognizes, it will accept the server certificate and establish the encrypted connection with no complaints.

NOTE

You cannot use name-based virtual hosting with separate SSL certificates. This is a limitation of the SSL protocol. The reason is that the SSL connection is established before HTTP. Since the machine, before HTTP, is referenced by IP address only, there is no host header that the server can identify a separate virtual host with. If you want to run multiple virtual

hosts with separate SSL certificates from one machine, you must use IP-based virtual hosting with a separate IP address for each host that you want to SSL enable.

The installation of `mod_ssl` added all the necessary configuration options to the `/etc/httpd/conf.d/ssl.conf` file that will enable SSL in Apache. Simply copy it to the places defined in the `ssl.conf` file and restart Apache to make your new certificate active.

```
$ # overwrite the default server key
$ cp acmewidgets.key ssl.key/server.key
$ # overwrite the default server crt
$ cp acmewidgets.crt ssl.crt/server.crt
$ /etc/init.d/httpd restart
```

Your new server certificate should be active and viewable through any `https://` URL on your server.

Apache will allow you to turn SSL support on or off for the Main server configuration, or VirtualHosts, by adding the *SSLEngine* directive, but will only allow you to use one certificate per IP address (see the Note above). You also need to add a *Listen* directive to tell Apache to listen on the registered port for HTTPS communication. The `ssl.conf` file should have added this. Let's take a look. The following example is from `ssl.conf` with comments and directives that we've already covered removed.

```
# load the module, listen on port 443, and register the MIME-types
LoadModule ssl_module modules/mod_ssl.so
Listen 443
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

# SSL handling options
SSLPassPhraseDialog builtin
SSLSessionCache          shmcb:/var/cache/mod_ssl/scache(512000)
SSLSessionCacheTimeout  300
SSLMutex default
SSLRandomSeed startup file:/dev/urandom 256
SSLRandomSeed connect builtin
```



```
SSLCryptoDevice builtin

<VirtualHost _default_:443>
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn
SSLEngine on
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key
</VirtualHost>
```

This is the minimum configuration for SSL on Apache. After installing your certificate, if you leave everything as-is in the `ssl.conf` file, then Apache will be waiting for requests on port 80 and port 443. If you want to add another HTTPS `VirtualHost`, then you can do it here. At the beginning of this example configuration there are a number of `Main` section configurations that will define some handling options for Apache's use of SSL — the module is loaded, and Apache is told to listen on port 443 for requests. New mime types are registered so Apache knows how to serve `.crt` and `.crl` files. Then caching options and more defaults are set. The `mod_ssl` developers enable SSL in a `VirtualHost` for organizational purposes, but you can just as easily put it in the `Main` section configuration file. Since this `VirtualHost` now has *DocumentRoot*, it will be inherited from the main configuration file and all files in the default web space will be served identically as before, but will be encrypted if you access them through the `https://` URL schema.

Modules

Apache's modules may be turned on or off as needed in the `httpd.conf` file. By default, Apache comes with many modules — the basic set is just about all you'll need to do anything we've done so far. But if you are trying to set up a particular service, WebDav or LDAP (Lightweight Directory Access Protocol) support, for example, then you probably need to uncomment that module and set configuration options that that module provides.

When we installed the `mod_ssl` package, it added a new configuration file for us that loaded the `mod_ssl` module. When this module is loaded all of the above `SSL_something` directives became available to us to use in the configuration, after

the module was inserted. Setting up other modules is very much the same, but you will have different sets of new directives to use with each module. To interface our LDAP server, for example, we would add the following to our configuration file.

```
# load the mod_ldap and mod_auth_ldap
LoadModule ldap_module modules/mod_ldap.so
LoadModule auth_ldap_module modules/mod_auth_ldap.so

LDAPSharedCacheSize 200000
LDAPCacheEntries 1024
LDAPCacheTTL 600
LDAPOpCacheEntries 1024
LDAPOpCacheTTL 600

<Location /ldap-status>
    SetHandler ldap-status
    Order deny,allow
    Deny from all
    Allow from acmewidgets.com
    AuthLDAPEnabled on
    AuthLDAPURL ldap://127.0.0.1/dc=acmewidgets,dc=com?uid?one
    AuthLDAPAuthoritative on

    require valid-user
</Location>
```

Similarly, if we want to enable WebDAV support for our DocumentRoot with the mod_dav and mod_dav_fs modules, we would need to add something like the following to our configuration file.

```
LoadModule dav_module modules/mod_dav.so
LoadModule dav_fs_module modules/mod_dav_fs.so
DavLockDB /var/lock/DavLock

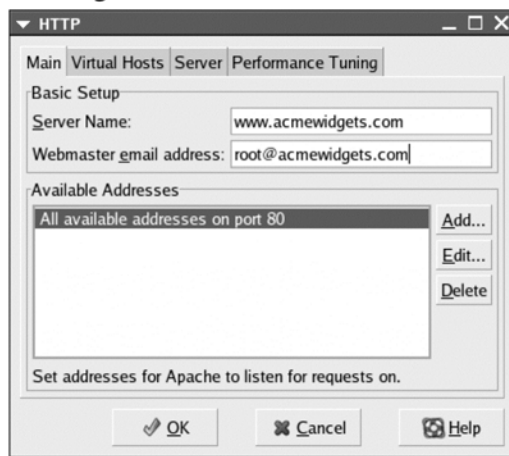
<Location /var/www/html>
    Dav On
    AuthType Basic
    AuthName DAV
    AuthUserFile user.passwd
```

```
<LimitExcept GET OPTIONS>
    require valid-suer
</LimitExcept>
</Location>
```

Graphical Tools

On RedHat versions 7.3 and later, there is a complete set of graphical server administration tools. These tools are all called *redhat-config-something* where *something* is the name of that particular service you want to configure. Some examples are *redhat-config-network*, *redhat-config-bind*, and *redhat-config-httpd*. Within the GNOME desktop, these tools are incorporated into the window manager and accessible through the **System Settings** menu. On Fedora Core systems, these configuration utilities changed their name to *system-config-something*. Examples are *system-config-network*, *system-config-bind*, and *system-config-httpd*. Regardless of window manager, you should be able to run these tools in X-Windows through a terminal by issuing the name of the command, as root. See Figure 9.11 for a screen shot of the HTTPD configuration tool.

Figure 9.11 HTTPD Configuration Tool



Though its looks may be a little deceiving, the HTTP configuration utility provided with Fedora systems, and accessed through the command *system-config-httpd*, is actually quite an elaborate and capable application. It is many times more granular than the IIS dialog box. Each tab has its own set of submenus and some

of them have submenus with their own tabs. By digging down into the submenus you can administrate everything we've covered in this chapter, all without touching the command line.

Another utility that will ease administration of Apache is WebMin (www.webmin.com/). This program will turn your Linux or UNIX server into a network appliance, completely administrable through a web browser. It not only has support for configuring Apache from a web browser, but it has support for configuring all of the server daemon programs typical to an Internet servers such as: Postfix, Bind/Named, MySQL, PostgreSQL, Sendmail, Procmal, SpamAssassin, to name only a few. It also has many configuration menus that deal with network behavior, bandwidth limiting, and TCP/IP Filtering with kernel-level Netfilter (Firewall). It's the Swiss Army knife of tools and a lot like Linuxconf for the Web. But that's not all. WebMin can also help you administer the underlying operating system. You can manage groups and users on the system, change passwords, manage filesystems, and you can view log files, edit config files, and manage boot-time options. WebMin has become a very powerful and robust utility for the system administrator. This highly recommended tool can be seen in Figure 9.12.

Figure 9.12 WebMin Administration Tool



Migrating Static Sites from IIS to Apache

Now that we have our two working Web servers set up and answering requests, migrating from IIS to Apache is just a matter of copying our documents from one to the other and renaming default index pages. Our site isn't currently dynamic, with the exception of the ASP hello world, but even that can be converted to equivalent PHP code. To transfer the files you can use any number of File Transfer Protocols. Since we always suggest avoiding clear-text transfer protocols wherever possible, we used OpenSSH and the WinSCP client to transfer our files from Windows to Linux. WinSCP is an excellent, open source, free software, graphical secure copy program that behaves a lot like any other FTP program and communicates with SSH or SFTP servers. See <http://winscp.sourceforge.net/eng> for more information about WinSCP and downloads.

After copying your Default.htm and Default.asp pages to your Linux server's DocumentRoot directory, rename them to index.html and index.php, respectively. The index.html page will work right away without modification and the index.php page should be modified as follows.

```
<!-- sample index.php -->
<html>
  <body>
    <p>Hello world.</p>
    <p>The time is: <?php echo date("h:i:s A"); ?></p>
  </body>
</html>
```

Summary

In this chapter we covered a lot of history, as well as a lot of territory, regarding IIS and Apache – the two most prominent Web server daemons on the Internet. By now you should be able to turn on the Internet Information Server, set up the default server with a default index page, configure virtual directories and virtual servers, as well as enabling SSL encryption. Configuring Apache, you may have found, is completely different, but the same facilities or equivalents are available, plus many more. You should be able to turn on the Apache Web Server, configure its default index page, Main server configuration, aliases and ScriptAliases, VirtualHosts, and SSL support. With this general knowledge at hand you should be able to install any number of modules to suit the needs of your organization. Additionally, the way to migrate a site from one server to another should be simple and clear – all that is needed is a good file transfer client.

Solutions Fast Track

Background: HyperText Transfer Protocol

- ☑ Tim Berners-Lee designed the HTTP protocol while at CERN in 1990.
- ☑ HTTP protocol built off ideas from early Gopher and Ted Nelson's idea of "hypertext" from his work with Xanadu.
- ☑ HTTP and the World Wide Web were very successful. In the early 1990s Microsoft and Netscape Communications started building their own Web servers and Web browsers.

Understanding Microsoft's Internet Information Server

- ☑ IIS is available in Microsoft's line of NT products, but some features are disabled in non-Server flavors. Configuration and administration of IIS is done through the Internet Information Services dialogue.
- ☑ IIS has support for virtual directories, virtual servers, and authentication, but if you want to run SSL, you need to purchase a signature from a Certificate Signing Authority before turning it on.

Understanding Apache Web Server

- ☑ Apache was built out of a series of code patches against CERN's early httpd program.
- ☑ Configuration of Apache is traditionally done through the text file `httpd.conf`, but there are also GUI tools and Web applications available to the system administrator.
- ☑ Apache has support for aliases, virtual hosts, authentication, and SSL, but due to SSL protocol limitations, you cannot run more than one SSL virtual host with name-based virtual hosting – you must use IP-based for multiple SSL certificates on one server.
- ☑ Apache has many modules, for expanding features, packaged by default.

Migrating Static Sites from IIS to Apache

- ☑ Copy files from Windows server to Linux using a file transfer protocol.
- ☑ Place files in the `DocumentRoot` directory.
- ☑ Rename `Default.htm` to `index.html`.
- ☑ Simple ASP file functionality can be achieved by using the freely available PHP language instead.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q. Where is the default Web directory in IIS?

A. `c:\inetpub\wwwroot`

Q. Where is the DocumentRoot (default Web directory) in Apache?

A. `/var/www/html`

Q. Where are the logs for Apache?

A. `/var/log/httpd` or `/etc/httpd/logs`

Q. What’s a quick test to check that the Web server is running?

A. At the command line, Telnet to port 80 on the local machine, then type in **GET / HTTP/1.0**

Q. What’s a quick test to make sure that my SSL certificate is working?

A. Just like the Telnet test on port 80, you can use openssl to test that port 443 is working correctly:

```
$ openssl s_client -connect localhost:443 -state -debug
GET / HTTP/1.0
```

Q. How do I test to make sure that PHP is working?

A. Under DocumentRoot, create a file called `info.php` and add the following to it. Then open `http://localhost/info.php` in a Web browser.

```
<?php phpinfo(); ?>
```

Q. What’s an RFC?

A. RFC stands for Request For Comments. RFCs are open Internet standards that define how all TCP/IP protocols work. They are in the form of white

papers that lay out best practices for developers when implementing a protocol. They are open in that they allow for people to make comments on them, change them, and update them as needed. The HTTP RFC, for example, has been rewritten many times to allow for new features to be implemented. There is an RFC available for every popular Internet protocol as well not-so-popular ones, like HTCPCP 1.0 (Hyper Text Coffee Pot Control Protocol). Please see the Internet Engineering Task Force's website at <http://ietf.org> for more information about RFCs.

- Q.** Where can I find out more information about Secure Sockets Layer protocol?
- A.** For theoretical overviews of how SSL works check out the OpenSSL web page www.openssl.org/ Apache's SSL/TLS Encryption page at <http://httpd.apache.org/docs-2.0/ssl/>.
- Q.** Who do I call when all this breaks?
- A.** Most popular distributions have support avenues you can pursue. Of course support generally costs money, but what you get is usually worth the price. Most Linux technicians are eager to help you get to where you're trying to go if you explain to them what you're doing. You can also search on the Internet for independent consultants who will be more than happy to help you implement your web services, for a modest fee. These services, it could be said, are where Linux gets expensive, but our experience has been that it's still less than vendor lock-in, in extreme examples, and balances out by building a professional relationship with someone who cares about your business's growth.

The Internet is an excellent place to find how-tos and FAQs for all things Linux and since the Free Software movement is just as much about culture as it is about freedom, Linux User Groups (LUGs) are an awesome place to get assistance and learn about managing your server. Do a search, cross-referencing your area and you're sure to find one in your area. If you don't find one, start one up yourself. The Linux Documentation Project has a how-to all about Linux User Groups at: www.tldp.org/HOWTO/User-Group-HOWTO-3.html#ss3.1.

Desktop Migration Roadmap

Solutions in this Chapter:

- Assessing the Current Desktop Environment
- Designing the Linux Desktop
- Testing the Linux Desktop
- Migrating Application Data and Profiles
- Training the Desktop Users
- Deploying the Linux Desktops

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

The process of migrating your Windows desktop users to Linux is similar to what we did in Chapter 1, migrating from Windows servers to Linux, but extra care is needed regarding the user's productivity – you have to pay special attention to individual users' systems settings and training needs to ensure that the move is swift. The process is the same in that you will go through the steps of information gathering, designing, and deploying just like in Chapter 1, but since we have already decided where our desktop machines will be within the entire network topology, we can skip that part of design. In this chapter we will analyze the Acme Widgets desktop computing environment and establish a roadmap for migrating their users to Linux with the Fedora Core operating system.

The process described here is meant to be a general overview of what is involved in the desktop migration process. We will be covering a relatively small in-house network like Acme Widgets, though the process described can be scaled upward to as many users and systems as needed. Throughout the chapter we'll offer up some suggestions to help tailor the migration roadmap to suit your company.

A successful Linux migration involves planning, a few extra computers, and some help: assessing the current software and hardware and designing the new environment are prerequisites to implementation. The information you gather will allow you to determine the best suitable alternative for the types of users in your company and develop a timetable for rolling out these alternatives to them. This is why we divided the migration process into the following stages: assessment, design, testing, migration, training, and deploying. These stages may serve as milestones on the migration roadmap, although they don't necessarily have to be done in order. Let's take a look at each of these stages.

Assessment involves assessing the various desktop environments and preparing reference documents. To do this we create a list of software applications, hardware, and licenses for each machine. This is called an *asset list*. By looking at the asset list we will be able to identify user profile types and create a *functional requirement specification* for the replacement desktop computers. The functional requirement specification is usually a spreadsheet or table of application resources each user type needs at his or her system. Assessment ends when we have these two documents completed.

In the design stage, we take the asset list and the functional requirement specification from the previous stage and create a *functional replacement specification* for

each user profile type. The replacement specification can also be thought of as an assignment of a Linux profile to the user profiles garnered from the asset list. We will use this specification to build the test machines and training computers that will be used later.

The testing stage makes sure that the test operating system replacement lives up to the replacement specification and meets the requirement specification from the previous step. Several checklist items should be tested to ensure that your users won't encounter an easily caught bug.

The migration stage can be done alongside the training stage if you have the IT staff to do it. While one person is teaching your users a basic introduction to the Linux, your IT staff will be backing up those users' computers and installing the new specified operating system profiles on them. To back up you'll need a network file server or CD burner. The installation can be done a few different ways –network installs, cloning hard disks, Kickstart methods, or general CD-ROM installs are all popular and easy methods.

Training usually requires that you have some extra hardware lying around. Depending on the size of the company, your resources will vary but you'll want to reserve some percentage of extra machines to be used as training computers. The easiest way to learn Linux is through hands-on practice. Having training computers will not only give your users time to play around with Linux, but will also allow you to test your operating system design in a real-world scenario. Obviously the more training computers the better, but a general figure would be seven to fourteen for a medium-sized company of about 50-100 users. In a small company like Acme Widgets, the training can be done one-on-one with the migration of the user's machine transpiring during after-office hours. The time the training will take can be as little as one day or stretched out longer over a week, but must be very elaborate – many operations on the Linux desktop are just like on the Windows desktop. The training will need to cover navigating the file system, launching applications, and setting up e-mail and preferences.

The final stage, deploying, involves answering questions and helping users adjust to the new environment as issues arise. Writing documentation to help recreate your desktop environment and short how-tos that explain users how to set up common applications and options can be incredibly beneficial. Part of the deployment also involves a bit of hand-holding while helping users configure e-mail settings, instant messaging, and groupware applications, as well as managing their backup directories on the file server and importing things like bookmarks from the old browser settings.

All of this assumes an able IT staff, but it can be done if you take the time to do it right.

Assessing the Current Desktop Environment

You have to know what you're working with before you can decide how to migrate. Take inventory of the kinds of users, the hardware and software, and the licensed software applications currently in use. With this information, you'll be able to determine if there is a Linux software solution, if the hardware is compatible with Linux, and what special requirements your desktop users may have. The next two sections will cover the various types of computer users you may encounter and how to take account of their desktop environments. With this information we'll create the *functional requirements specification*, which defines what applications and hardware demands, if any, those users have.

User Types

In an office computing environment users generally fall into a number of well-known categories. In a small organization, this categorization may not be of critical importance, particularly if the computer use by employees is generalized. In a larger organization with highly specialized workers, it is useful to identify the number and types of computer users so that you can design a replacement operating system for them. General categories are the following.

Kiosk

Kiosks are simple workstations with very few applications. Many kiosks employ a non-standard form-factor such as an embedded system or appliance. Kiosks are frequently used for point of sale, information presentation, web browsing, and custom applications. In some cases, employees use kiosks – in other cases, business clients or patrons may use kiosks. Internet lounge systems and information presentation systems are also examples of kiosks.

Basic Knowledge Worker

A basic knowledge worker primarily uses general office applications and little else. This typically includes a web browser, an e-mail/PIM client, and an office suite featuring word processing, spreadsheet, presentation, drawing, and possibly

database software. Basic knowledge workers use no specialty applications, but may perform some transactional type of work. Typical positions that fall under the basic knowledge category are data entry clerks, transcriptionists, and administrative assistants.

Transactional Worker

This is a fairly general category that features several subtypes. A transactional worker is someone who primarily performs nearly identical well-defined tasks and sometimes uses specialty software applications to perform these tasks. A transactional worker typically holds positions such as help desk representative, order entry clerk, shipping and receiving service representative, or in a support center interface.

Technical Worker

Technical workers, such as software developers, CAD/CAM/CAE hardware developers, and systems or network administrators typically use their desktops to design, develop, or use special development applications like Integrated Development Environments (IDE). They generally require the same applications as the basic knowledge worker, such as e-mail, web browsing, and office suite applications.

Advanced Knowledge Worker

These are the power users who have in-depth knowledge of office applications and know how to use advanced features and functions. They are likely to be sophisticated users with many add-on devices and peripherals as well as additional software to make them more efficient. An advanced knowledge worker can be found in a wide range of positions, from an executive assistant, to a recreational computer geek. In small companies, advanced knowledge workers may perform system administration or other IT tasks.

Creating the Desktop Asset List

Performing a proper assessment of the current desktop environment is the most critical part of the migration. The information gathered in this stage will be referenced over and over again throughout the migration process. A *desktop asset list* may be prepared to help analyze your user's requirements. This document is an inventory of the hardware and software on the desktop computers. Assessing the

hardware is just a matter of logging speed, capacities, and components/peripherals of the desktop machines. Assessing the user's software environment involves recording the desktop software in use, software licenses, and sometimes even the configuration of the software. We'll discuss that last one more thoroughly in Chapter 11.

Most organizations will already have some type of asset list or inventory list from the procurement of their hardware. If you do not already have one, this document is easily prepared. The items tracked on it are typically: serial number; computer name, make, and model; CPU make, model, and speed (Mghz); and RAM and disk capacity. It will suffice to say that you will want to gather all the particular information about the major parts of the computer, as well as any attached peripherals, including add-in cards and their makes and model numbers. Compiling this information in one place will greatly expedite the process of determining Linux hardware compatibility and will form the basis of our replacement specification. Table 10.1 illustrates what a very basic hardware/software/license asset list may look like.

Table 10.1 Example Asset Listing for Acme Widgets

Serial # Computer Name, Make, Model	CPU Make, Model, Speed, RAM, Hard Disk, CD	NIC, Sound, Video, Modem, WiFi, Other	Printer, Scanner, USB External Drives	Operating System, Software	Licensed Applications
1001 LPTP_1001 HP Omnibook 6000	Intel PIII, 850Mghz, 256M, 20Gig, DVD/CD-RW	3com 10/100 "mini PCI" NIC and 56K modem, ATI Rage 8Meg, ESS Meastro, Orinoco 11mbps WiFi PCMCIA,	HP Photosmart 1 200 USB, Generic USB 20Gig storage, 256M MP3 player (USB storage)	Windows 2000 Pro, Graphic Design, Office App, Email, Web, Web Development	Windows 2000 Pro, Office 2000, Photoshop 5, Ultimate FTP, Dreamweaver
1002 LPTP_1002 Panasonic Toughbook CF-71	Intel PII, 300Mghz, 192M, 20Gig, CD-ROM	Neomagic Gen. SB, Neomagic 2Meg video, Aironet 11mbps WiFi PCMCIA,	Kodak DC 240 USB camera	Windows 98 SE, Graphic Design, Word Processing, Email, Web	Windows 98 SE, Photoshop 5, Word 6
1003 DKTP_1003 PC Compatable Desktop, Vendor Unknown	Dual PIII 700Mghz, 512M, 60 Gig+ 40Gig external, CD-RW	2 Lite-on 10/100 NIC, NVIDIA Riva TNT 2, Trident SCSI	External Seagate 40Gig SCSI Hard Disk, USB Mouse/KB	Windows 2000 Pro, Graphic Design, Office App, Email, Web, Web Development	Windows 2000 Pro, Office 2000, Photoshop 5, Ultimate FTP, Dreamweaver

As you will come to learn, the more specific the information about the hardware the better when it comes to determining Linux compatibility. Most of the information above was gathered right from the Device Manager in Windows, but it is wise to actually inspect the hardware since names displayed in the Device Manager are often different than the actual model of the card. If it is possible, open up the case of the computer and write down make and model number (if one is available) of add-in cards and motherboard. You may already have this information on purchase order receipts. This will help resolve ambiguities arising from multiple model types with the same branding name.

Next, determine which applications are being used in your organization and what functionality they are providing. Try to build as detailed a picture as possible of what actual tasks your desktop users are performing. This information plays a later role in shaping the desktop requirements.

Kiosk users, basic knowledge workers, and transactional workers typically use an office suite, an e-mail program, a web browser, and possibly a few specialty programs like graphics manipulation. These requirements are easily satisfied by open source applications, many of which run on both Windows and Linux. To run office-like applications, use OpenOffice.org. To do web and e-mail, use Mozilla. To run image editing, use The GIMP. These alternative solutions today are complete applications: they have an intuitive GUI, are feature-rich, and integrate with the window manager. We know that OpenOffice.org, from <http://openoffice.org>, provides a copyleft office suite for free. It can handle Word/Office XP/2000 .doc files, spreadsheets, presentations, drawings, HTML (Hypertext Markup Language), and more. Mozilla, from <http://mozilla.org>, is a robust web browser, email client, news client, IRC client, and more in one suite of tools. And we know that the Graphics Image Manipulation Program known as “GIMP”, from <http://gimp.org>, provides all of the open source photo editing power anyone would need.

Advanced knowledge workers and technical workers will often have some of these legacy applications lying around and therefore, it follows that they will have more complicated application requirements as well. These desktops will require more time determining appropriate Linux desktop solutions. In some cases, an emulator may be required to give them the same functionality on the Linux desktop as on the Windows desktop. However, one advantage while migrating these users is that they are typically the most willing to accept change, and some of these users may already be experimenting with alternative solutions to their proprietary woes.

We can see from our asset list for Acme Widgets that their users primarily consist of a technical worker and two advanced knowledge workers.

Cataloging File Formats

It's also good to know what kind of file formats your users require. Determine the file formats being used in your organization by opening up applications on the desktops and trying to read from or write to files. Most programs will have a pull-down menu in the **Open File** or **Save File** dialog boxes that will allow you to see what kinds of file formats that program expects. Many of the common file types (such as .doc, .xls, .ppt, and .html) are easily handled in Linux by open source applications such as OpenOffice.org and Mozilla. Make a list of all of the application-related file formats in your organization to help when searching for a replacement application. Table 10.2 lists some common file formats with the corresponding Windows and Linux applications that read and write to those formats.

Table 10.2 Common File Formats

File Type	Extension	Windows Application	Linux Application
Hypertext	.htm .html	Internet Explorer	Mozilla, Firefox, Konqueror
Document	.doc .rtf	MS-Word	OpenOffice.org Writer, Abiword
Spreadsheet	.xls	MS-Excel	OpenOffice.org Calc, Gnumeric
Presentation	.ppt	MS-PowerPoint	OpenOffice.org Impress
Personal Store	.pst	MS-Outlook	None that work directly with .pst
Images jpg .gif .bmp .png	Photoshop	Gimp	
Audio/Video	.wav,.mp3,.mpg	Windows Media Player	X Multi-Media System (XMMS)

Functional Requirement Specification

The *functional requirements document* will form the basis for choosing the applications to use in Linux. This document is basically a reiteration of the software components of the asset list, but leaves out the applications that are not needed. A project or IT manager who can make executive decisions usually creates this document and it frequently serves as a basis for Request For Proposal (RFP) documents when outsourcing solutions. A functional requirements document will also serve to achieve buy-in from affected users and ensure that their needs are met. Review this document with personnel while they are using their computers to make sure nothing is missing – simply digging through **Start Menu | Programs** is not sufficient. If time permits, browse the hard drive and make sure there are no surprises. Spending adequate time evaluating and documenting your findings at this stage is a key ingredient for a successful migration. Table 10.3 has an example of a simplified functional requirements document for Acme Widgets.

Determined from the asset list and considering the file formats from the previous step, the following functionality is required of the computers at Acme Widgets main office.

Table 10.3 Functional Requirements Document

Application	Function
Office Suite	Must support MS Document, Rich Text, Spreadsheet, and Presentations
Web Browsing	Must support HTML 4.0, CSS 2.0, Frames, Javascript
Web Development	WYSIWYG HTML Editor
Email	Must support encrypted IMAP, POP, and news protocols. Must handle folders, calendaring, and handle HTML formatted mail
Graphics	Must support TIF, PNG, GIF, JPEG, Postscript, Bitmap, and PSD image formats
Media	Must support WAV, MPG, MP3, and OGG media file formats

Designing the Linux Desktop

With the assessment of the Windows desktops complete you can begin designing the replacement Linux desktop solution. The design, naturally, will depend heavily on the functional requirements and therefore the *functional replacement document* created in this section provides solutions for the requirements specified in the assessment stage. This document can then be incorporated into a proposal (when outsourced), for example, or sent back to the project manager from the IT staff as solutions to her problems. Using the functional requirements document as a guideline, determine the suitable Linux application that will satisfy each requirement for each type of worker in your organization. This information will be incorporated into the functional replacement specification.

Designing & Planning

Asset Lists

Some aspects of the migration may require an order that you will need to follow, while others may be simple and straightforward. Preparing your asset list will help determine ordering of tasks to perform during the migration.

For most major applications like in our requirements example, we already know what software we want to use to replace common programs, but with some legacy applications it may be necessary to search the Web for a suitable replacement. Freshmeat.net (<http://freshmeat.net>) is a community-adopted standard repository of open source and free software projects. Search through there and in many cases you will be surprised that the open source community has already made the application that you just can't live without.

In some instances where an employee is specifically trained to work with one particular piece of software, it may make sense to continue using the proprietary application. These programs can often be run in Win32-emulating environments from within Linux. These Windows *work-alike* alternative solutions vary in complexity and price. Some of them simply provide the libraries necessary to run Windows applications from within Linux, and others create a *virtual machine* that

allows you to install a licensed copy of Windows operating system inside of it. If you find yourself needing one of these kinds of solutions, look into CrossOver Office, WINE, VMWare, or Bochs. A search on Google or Freshmeat will turn up any of these projects.

For particularly stubborn users who don't want to switch operating systems, it is possible to ease them into it by supplying them with alternative software while they are still on Windows. OpenOffice.org, Mozilla, and the GIMP are all available on Windows today, for free. This kind of in-between solution will allow for a properly staged migration and will ease the learning curve since the user will already be familiar with some of the desktop software before they begin with Linux.

Some vendors also provide Linux versions of their Windows software. If your company has purchased a license for a particular application that is supported by the vendor in Linux, they may be willing to transfer the license. In some cases, Linux binaries may already be packaged on the installation CD-ROM.

For any other Windows application that you will want to retain, you have to use one of the Windows work-alike solutions, as we already mentioned. Generally applications that are simple and run in older versions of Windows (such as Windows 98) usually work fine with WINE, which provides native libraries for Windows applications in Linux, for free. Just issue the **wine some.exe** command in Linux. Larger applications that require elaborate local installation, like MS Office 2000, will require CrossOver Office or installation into a virtual machine. Since Acme Widgets' requirements are easily provided for with common Linux packages, we won't be getting into any greater detail about hybrid solutions. See chapter 11, "Inside the Linux Desktop", for additional details about Linux desktop environments and common applications.

Acme Widgets' user profiles will require two different Linux operating system solutions. For the basic knowledge worker, we'll want a basic desktop installation of Linux, paired down to run smoothly on the older hardware. For the advanced knowledge workers, we'll need a developmental install of Linux that provides libraries and development tools like IDEs to the user. The Linux installation program packaged with the distribution will provide options to select these types of operating system profiles as well as even more granular preferences. Any trained help desk or technical support engineer will be able to install a Linux operating system on a computer that meets these descriptions. These operating system profiles will be used in making our functional replacement specification.

Tools and Traps

Linux Terminal Services

Another option for Linux desktops, suitable for larger network installations with many kiosk and transactional workers, is moving to a Linux Terminal Services model. Linux Terminal Services is a way to export desktop environments from a central server to *thin clients* on the internal network. This methodology scales very quickly and is more easily administered than individual desktop machines. Its drawbacks are that it's not really applicable to small networks and it is more complicated to implement. In addition, the network becomes a dependency in this model and you will usually want to have one very fast machine with a lot of storage.

If you do elect to deploy Linux Terminal Services, it is a good idea to build in redundancy on the terminal server by pairing it with a failover machine and making regular backups. The Linux Terminal Services model becomes drastically cheaper with each system you add in terms of hardware required per station and ongoing maintenance. This model scales well since adding another user to the network is simply creating another account on the terminal server and plugging in another thin client to the network. However, also keep in mind that designing a Linux Terminal Services solution requires a considerable time investment in planning and testing to get it right. However, the value is that you have one machine to focus administration. Linux Terminal Services is the best bet for manageability, but may not be possible or desirable in all cases – most notably, small ad-hoc networks. Please check out the Linux Terminal Services Project home page for more information (<http://ltsp.org>).

Functional Replacement Specification

Next we'll create the functional replacement specification and the functional requirements document. The latter document basically maps a Linux program back to the functionality that has to be replaced. You can create a replacement document for each user profile in large installations, but for our purposes it will suffice to say that our advanced knowledge workers have the same configuration as the basic knowledge workers, plus additional development tools. An example of this document is in Table 10.4.

Table 10.4 Functional Replacement Document

User Type/Operating System Profile	Application	Linux Program
Basic Knowledge Worker/Desktop	Office Suite	OpenOffice.org
Basic Knowledge Worker/Desktop	Web Browsing	Mozilla
Basic Knowledge Worker/Desktop	Email	Mozilla
Basic Knowledge Worker/Desktop	Graphics	The GIMP
Basic Knowledge Worker/Desktop	Media	XMMS
Advanced Knowledge Worker/Development	Web Development	Quanta, Scream

Since Linux distributions are nothing if not robust, all of these replacement programs are already packaged with popular distributions, including the Web development IDEs. Making them available to the user is just a matter of choosing them during installation.

Building the Training Computers

If you are a small organization with only a few users and computers, you can probably skip this step. For larger organizations that need to roll out dozens of desktop machines, you will want to create prototypes of your operating system profiles on some extra training computers. This will serve the triple purpose of allowing the IT manager to specify an installation methodology for imaging all those machines, providing a place to test those system profiles, and providing a sandbox to allow your users to play in. Mirroring hard disks, creating installation profiles with Red Hat's Kickstart, network installations, and CD-ROM installations are all options available with any popular Linux distribution. Look into these options if you have to provide for half a dozen or more desktops. For Acme Widgets we'll just use the CD-ROM installation method and then make a Kickstart disk after the installation in case we need to image another similar machine for the company in the future. A search on Google will turn up many examples for issuing Kickstart installs.

Testing the Linux Desktop

Once the training computers are built you can test them by designing a test plan like the one in Chapter 1. The test plan should ensure that the system profiles meet the replacement specification. The test plan should ensure that all the software we want for that user profile is installed and runs without requiring extra libraries or packages that weren't installed during installation. It should check that the default window manager is intuitive and easy to use – you can also customize the window manager to the needs of that user profile by making the most commonly used programs and applications available in the tool bar, for example. We'll be discussing this more thoroughly in the next chapter so we won't go too far into the details now.

When you are satisfied that the installation meets the the system profiles, write down the steps you took to get it that way and repeat the process across your other systems. This is the simplest way for dealing with just a few computers. If you have many computers to install, you can use a Kickstart method, as already mentioned, or a disk mirroring method to image the other computers. With a small network like the one we're working with, it's quite easy to do the same installation three times.

We'll revisit testing and test plans in more detail when we migrate our users' data to the new Linux machines.

Migrating Application Data and Profiles

Migration, at this point, is just a matter of exporting and backing up the user's data, installing Linux, and then importing and copying the user's data back to the new Linux system. After assessing, designing, and testing the design, it should be pretty clear what needs to be backed up and which applications will be available on the Linux desktop. The Linux operating system profiles are tested and we know roughly how long it should take to install one of them to a machine. Beginning the next stage is just a matter of scheduling with the users.

The time it will take to install Linux on your users' machines will vary depending upon the skill of the technician, but generally, on modern hardware (around 1 Ghz processor/256Megs of RAM), it shouldn't take longer than 2-4 hours per machine. This time will be reduced as you add more machines, since you can do concurrent installations by burning several CD-ROM sets of the distribution. This installation time will also get shorter when you switch to other installation mediums, like network installs or Kickstart. Since there are no auto-

mated programs for dealing with cross-platform application migration, much of the process has to be done by hand. This is why some IT managers will opt to have technicians migrate their users' computers after business hours or on the weekends if possible.

Backing Up Desktop Systems

Simple backup options available to small- to mid-sized networks are network backups, USB hard disk backups, or CD-ROM(s) backup (if the host machine has a CD-RW drive). Each method has different benefits depending upon the size of the network and resources available. If you have enough storage capacity on a file server that can cover all your users' data, then a network backup is probably the quickest. External USB hard disk backup is probably the easiest since it doesn't involve having other services running. CD-ROM backup is probably the cheapest as far as media is concerned, but has the limitation of only being able to store 650 Megabytes of information on one disk, so backing up even one machine may take many disks. DVD backups are another option, but this requires a DVD-RAM drive and media that can get expensive and slow. Given the capacities and prices of hard disks, network backup, or USB hard disks, these backup types are probably the best solutions. For larger networks with many desktop machines, a file server is not just recommended, but is necessary.

Although backing up user mail, address books, and bookmarks will vary slightly for each application, all the user data as well as Windows-formatted versions of those resources for a user will be kept in a user's directory under C:\Documents and Settings. It is therefore usually sufficient to back up home directories and be able to reinstate them should the need arise for a rollback to Windows. Before you perform the backup, visit each common application on that user's desktop and export mail, address books, and bookmarks to a text-based format by using the export feature provided in that program. Save these exported files to the user's Desktop. You can also ask your users to do this for you by providing them with short instructions in an memo.

The Documents and Settings folder contains a sub-folder for each user on the system. Inspecting the properties of each sub-folder will show its size. Back the folders up to your storage media. As we've already discussed, there is a lot of flexibility here and will depend on your setting and resources. Also back up the All Users directory. With older systems, like Windows 95/98, in addition to backing up user directories, you'll also need to check the C: drive and any other physical drives or partitions for stray configuration folders. On NT systems and

newer you shouldn't find any, since it is a multi-user platform, but on older Windows 95/98 computers, which are not multi-user systems, it is common for programs to save data to their program directories under C:\Program Files.

By now, inside each of the backed-up user folders, you should have all the mail, bookmarks, address books, profile settings, and user data you need to migrate a user to Linux and back again.

Installing Linux

Installing Linux, as mentioned earlier, is very straightforward. You can do network installs, CD-ROM installs, or mirror hard disks from your testing computers. For Acme Widgets' computers we are installing straight from CD-ROM with Fedora Core Linux. Fedora can be downloaded from <http://fedora.redhat.com>. Two systems will get a Development installation of packages and the third will be a Desktop. After installation, we'll install **apt for rpm** on each, available from <http://freshrpms.net/apt>, and run `apt-get update`; `apt-get upgrade`; `apt-get dist-upgrade` to update packages to their most recent release.

Importing User Profiles and Preferences

Once the installation is complete, create the user accounts and login as the user. Mount the network file shares, USB drive, or CD-ROM media and start copying user files to their Linux user directory. From there, open each application and follow the importing procedures for that application. For example, to import an exported mbox file into Mozilla, close Mozilla, and copy the file to `<mozilla_userdir>/Mail/Local Folders`. It will then be available under Local Folders inside Mozilla Mail. The next chapter will cover this in greater detail, but you get the idea.

Training the Desktop Users

Training should cover navigating the desktop, opening common applications, and mounting CD-ROMs. Linux desktops today are very robust and intuitive; it is like switching to any other operating system. Opening a command line and teaching the user a few basic UNIX commands like **cd**, **pwd**, **ls**, **cp**, **mkdir**, and **rm** is probably the fastest way to orient a new user to the file system. Next, open up Nautilus and show users how to navigate the file system graphically.

Training Guidelines

There really is no set of guidelines for training. It depends on the aptitude of your users. Encourage them to take notes and ask questions. Show them how to find answers in documentation provided with the distribution and online through search engines like <http://google.com/linux> and repositories such as the Linux Documentation project at <http://linuxdoc.org>.

Basic Linux desktop training can be done in one day with a group and an instructor hired from a training consultancy group, or in many instances the technician doing the install can spend a few hours with the user and get him or her oriented and up to speed. Undoubtedly, your organization's needs will vary and you have flexibility here to be assessed when the time comes.

With a small network, one-on-one training is feasible. In larger networks you will want to schedule group trainings with an instructor while the IT staff is busy backing up and re-imaging computers. You can schedule this on a Friday and allow the IT staff to work out bugs over the weekend and provide the users with new systems on Monday, minimizing down-time. There are lots of possibilities available, but the big thing is to empower your users by providing them with the ability to discover answers on their own. A little bit of encouragement will go a long way here and you'll see users taking to Linux effortlessly.

In some cases, training will be required. In others it might not be needed at all. Migration can be spread out over a larger period of time if your organization demands. In a small enough organization you might be able to just have someone on hand the day new applications are rolled in. All of this should be documented in your migration plans and will depend on your functional replacement document. The following are a few more hints to help you along the way.

Take the list of Linux applications you are going to deploy and make a matrix based on functionality, comparing the Windows equivalent to the Linux application. Find out what is the same and what is different. Quality documentation can be generated from this process and you are on your way to creating some of your course material for future training. Move through the simple tasks first and get the bulk of it done, spending the remainder of your time documenting exactly how you do the various processes you documented during the desktop assessment phase, taking screen shots along the way.

Never forget you are asking your users to change – generally people fight change – your job here is to make it painless. You can do that by evaluating, planning, and communicating your goals. Ask questions, especially when it has to

do with the business process. Spend the time to find the best way to make things still work smoothly and easily when using different applications.

If you are not able to do this or are having a hard time, you can search the Web for local Linux Users Groups (LUGs) and meetings. Educational/training houses might also have something on the various applications you are planning to deploy. You can also find trainers through LUGs. If you search around, you're sure to find an able hand.

Linux Desktop Differences

Most of the differences on the desktop between Windows and Linux are environmental. That is to say that they are underlying and mainly are behind the scenes. We'll be covering this more elaborately in the next chapter. The majority of differences, however, are cosmetic. In the two most popular Linux desktops, GNOME, and KDE, there is a desktop navigation bar with configurable start menu, task bar, and toolbar, and it can be configured to include quick-launch applications and applet utilities – many of which come by default with the distribution and can be turned on or off through Preferences. These desktops act and behave, for the most part, just like a Windows desktop. The file system is different from Windows, as we already mentioned, and will need special attention.

On Windows, home directories are stored in C:\Documents and Settings, while on Linux, the equivalent directory is /home. Since clicking on any file manager application from the Linux desktop will start the user in his or her own home directory, the /home/<username>/ directory is a good place to copy the remaining backup data. Hidden files that begin with a dot will still be hidden – most of them are configuration directories. You can dig through these and possibly import more settings from other applications. An example would be when you're setting up instant messaging clients.

Transactional workers will probably be the easiest to migrate and teach new applications to, as they use far fewer applications during the course of a workday and generally use fewer features of each. The toughest ones to migrate are likely to be advanced knowledge workers, as they not only use more applications, but they also use more features of each application. However, they are also generally the most willing to learn, so it balances out.

The attention applied to these details, again, will be driven by your technical requirements specification, functional requirements specification, and how they map back to your new applications. In some cases the differences might be huge,

in others they might not even be worth mentioning. Every migration will have different requirements.

The larger your deployment is the more sense it makes to stage everything. In a small office environment it's easy to install and customize each system to the user's needs, but once you have over 15 desktops, the documents we created earlier in this section become exceedingly beneficial.

Filesystem Differences

In Windows, C: is the top-most directory of the first hard drive partition. In Linux, this is represented by the forward slash /, commonly referred to as the *root* directory – *everything* else in Linux descends from here. To Linux, everything is a file. There are seven types of files in Linux, but users only need to know about device files, regular files, and directories. The first level under / contains system-wide folders that the administrator uses. The /home directory is where the home directories are stored, as mentioned earlier. The /dev directory contains device files for all the hardware and devices on the system. The /usr directory has shared user files, but many command line binaries available to the user will be in /bin. In Linux, in order to use a device like a CD-ROM drive, or external USB drive, you need to *mount* it to a particular spot in the file system. Nowadays, there are auto-mounting utilities that do this for you and they mount removable media in the /mnt directory. In Windows, you would access your CD-ROM drive through the D: directory, but in Linux you access it through /mnt/cdrom.

You can make migration easier on your users by creating familiar folder names and shortcuts for them on the desktop such as “Shortcut to My Documents” and by creating symbolic links to commonly used items.

Other Differences

Other aspects of Linux are a little more challenging to explain since there isn't an equivalent in Windows. Some examples would be virtual desktops, virtual consoles, and differences in window managers. Changing monitor resolution in Linux can be done on the fly and there are hot keys for restarting your X-Windows session as well. Using devices is a notion that may take some time getting used to, particularly USB devices. If your users will be using removable media, for example, you need to make sure you have autofs or automount setup and place a shortcut on the desktop so it's just a matter of plug-in and browse for them. A lot of this will be based on how your users use their systems. Different departments of the organization will also have different needs. Again, refer back

to your functional requirements specification to see if you need to create a shortcut for an external USB drive on that user's system.

TIP

Window Managers – In Linux, the X Windows System provides the windowing environment and window managers are an abstraction layer between the user and the windowing environment. There are many different window managers available for Linux.

Virtual Desktop – Most window managers in Linux have multiple desktop workspaces. You can flip between them by using the applet on the desktop menu bar.

Virtual Console – By pressing **CTRL + ALT + F2**, you can jump to a virtual console from the window manager and log in at a command line. There are more consoles on **F3** through **F6**. Pressing **ALT + F7** will get you back to the window manager again.

Alternative Application Equivalents

Windows users will be used to a certain set of applications for the various functions they perform. These functions may have a different name in Linux. The new naming convention is one of the harder pieces of knowledge to disseminate. With every name being a new one, the more applications you add the closer to overwhelming it may become. You could use Windows icons for the replacement applications to make it easier to associate the new names with the old. Creating an application matrix and storing it on the intranet will undoubtedly be useful in large organizations. A simple document can be maintained on a file share of these alternative equivalents and linked to from the user's desktops or put in the `/etc/skel` directory, which is similar to the All Users directory in Windows. Putting files in `/etc/skel` will propagate common user configuration files to all new users.

The wide variety of applications, all with new names, is one of the largest obstacles for anyone migrating to Linux from Windows. Linux installs many more applications and utilities onto your system than Windows. Clicking on the main menu will present you with a wide range of choices.

There are many full-featured applications and multiple options available for the most common tasks such as writing documents, using e-mail, browsing the Web, or doing development work or graphics, as we have already mentioned.

There are also countless games, configuration and monitoring tools, and scientific and diagramming programs. If you have installed apt, you can run **apt-get install synaptic** and allow your users to pick through a vast repository of applications.

Every Linux distribution can be set up using something similar to apt-get, which enables new applications to be installed just by typing a short command as root in a console window. There are several variants to apt-get, such as urpmi from Mandrake and yum from Red Hat. These utilities have no equal in Windows and greatly simplify the chore of setup, administration, and provisioning of new programs to your users.

Getting Help

Though there are some large commercial support institutions, you rarely need to call them unless you are working at the enterprise level yourself. Just about all your questions can be answered in the documentation provided with the distribution, on the Internet, or through other means like LUGs or consultants. Packaged with every distribution will “Help” options in most aspects of the GUI providing documentation to that interface. In addition, every packaged installed on the system will have installed their own documentation in **/usr/share/doc** or in **man** (manual) pages. You can type **man bash**, at the command line, for example, to pull up the manual page for the BASH shell. Further information about BASH is available in the **/usr/share/doc/bash-some_version** directory.

On the Internet you can search mailing lists, news groups, bulletin boards, and articles. Often, solving a problem is just a matter of finding a solution someone already came up with. If you’re just trying to do an ordinary task, it can save a lot of time just by looking on the Web for a solution someone already came up with. Similarly you can search for LUGs and consultants just by plugging your area and LUG into Google. Or your area and Linux consultant will turn up some hired help. In any event, help is out there when you need it.

Deploying the Linux Desktops

In a small organization deployment is just the process already described above of backing up the user’s computer, installing Linux, and returning the user’s data to the new desktop. In larger organizations these steps will have to be staged on the migration roadmap. Also, for larger organization, where one-on-one support is not an option, documenting all the processes involved is invaluable. And you should be able to predict many of the common problems your users will have.

When the users go back to their new systems, all the same functionality should be provided for as on their old systems, but much of it is categorized differently, named differently, and will naturally take a little bit of time to get used to. You will probably want to monitor your users for a little bit, retaining a technician to be on call for at least a week during the initial roll-out period to help with users' questions. Your users will probably need assistance managing printers and setting up old instant messaging accounts. Anticipating a few questions here will be helpful.

Documentation

Keep an FAQ of the questions you get from your users. We haven't talked much about documentation in this chapter because the functional requirements specification, the functional replacement specification, the asset list, and any application matrix you may have created all serve as excellent reference materials. For the users, you could create short and to-the-point how-tos for setting up mail accounts or IM accounts and put them in an easily accessible place on the network. You'd be amazed how much this could save in technical support. A trained, hired consultant will know to think of these (experience is likely to have taught them).

Summary

This has been an overview and introduction to some the “gotchas” you might encounter while on the road to migrating Windows desktops to Linux. We’ve presented what we’ve found to be best practices in staging the migration process so that it progresses smoothly. We’ve covered assessing the current desktop environment, designing and testing the new environment, migrating your users and training them, and finally, sending our new computers and new Linux users into a production environment. Of course, this roadmap, unlike an ordinary roadmap, may be altered to suit the needs of your organization – more like a recipe, perhaps. Following the steps presented here will ensure the most seamless and painless migration possible. It seems that all change incurs growing pains, but the pains can be minimized if you plan ahead. The user interfaces are quite polished now. Yes, this is a good time to switch.

This chapter should have served as a roadmap to migration. The next chapter we will elaborate more on the particulars of setting up the Linux desktop.

Solutions Fast Track

Assessing the Current Desktop Environment

- ☑ Take inventory by making a hardware/software/license asset list.
- ☑ Develop a functional requirements specification so you have a definite target moving forward.
- ☑ Determine your user types and catalog file formats to aid in design.

Designing the Linux Desktop

- ☑ Consider the network topology and user needs.
- ☑ Identify comparable solutions in Linux.
- ☑ Develop a functional replacement specification.
- ☑ Create operating system profiles and build training/testing computers with them.

Testing the Linux Desktop

- ☑ Follow a test plan to ensure your operating system profiles fully meet the requirements.
- ☑ Manually test commonly used applications.

Migrating Application Data and Profiles

- ☑ Schedule time with your users to work on their computers.
- ☑ Back up user data, install Linux, and copy user data to Linux.

Training the Desktop Users

- ☑ Schedule training with your users.
- ☑ Cover file system differences, interface differences, common applications, and basic command line functions.
- ☑ Instruct users on how to get help.

Deploying the Linux Desktops

- ☑ Document everything.
- ☑ Expect a lot of technical support questions for the first week after roll-out.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Where’s that application?

A: There are thousands of applications for Linux and as many places to find them. Try out apt repositories for searchable and readily installable software. www.linuxmafia.com/kb/Apps also is a great place to start.

Q: Where can I download new applications?

A: Freshmeat (www.freshmeat.net) and SourceForge (www.sf.net) are two great places to download new open source applications.

Q: Should I try to perform all of the desktop migration steps in one day?

A: Migrating everyone in one day can be difficult for users. While it may work, frequently the migration will not go exactly as planned. In most cases it is better to spread out the disruptions to users, and allow sufficient time to troubleshoot unexpected problems.

Q: What is the Return On Investment (ROI) with a Windows to Linux desktop migration?

A: That question is difficult to answer because there are many ways to calculate ROI, and some advantages of Linux cannot easily be measured in dollars. All migrations initially cost time and money, but successful migrations will save money over time in licensing fees and maintenance. For organizations facing costly proprietary software license purchases, migrating to Linux and open source is almost always the less-expensive option.

Inside the Typical Linux Desktop

Topics in this Chapter:

- Common Desktop Environments
 - X Window Servers, and Window Managers
 - E-Mail and Personal Information Management Clients
 - Web Browsers
 - Office Application Suites
 - Running Windows applications on Linux
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

When it comes to discussing Linux, many consultants and vendors focus on advantages that appeal to what we call the “propeller head.” The tech set focuses on selling points such as system stability, the possibility of enhanced security, and the fact that Linux can save companies money on licensing.

Most end users, however, are not very interested in these topics; they simply care about their desktop experience. Users state that they want a desktop that is “intuitive” and “easy to understand.” What they are implying, however, is that they want a desktop similar to what they already know. One of the things that Microsoft and Apple have done admirably is to convince end users that their interfaces have always intuitive and easy to use, even though these companies have changed these interfaces radically over the last 10 years.

Remember, your job will be to please the “average end user.” These end users will want to know how to access productivity applications once they have logged on. They will want to know how to locate files on the hard drive and open them with the proper application. They really tend to care very little about anything else.

So, in this chapter, you will learn about how to choose the proper desktop environment and window manager for your clients. You will also be able to recommend e-mail, Personal Information Management (PIM) and Web browser applications so that end users can migrate from applications such as Outlook, Outlook Express, and Internet Explorer. Because end users will want to open Word, PowerPoint, and Excel files, you will also need to recommend the ideal Linux office suite, as well as additional open-source solutions. By the end of this chapter, you will be able to ensure that employees will be able to remain productive on the desktop.

Common Desktop Environments

“So, Linux uses the command line, right? I mean, you don’t have a graphical environment like Microsoft Windows, do you?” Because you are familiar with Linux, you might think that this is a silly question. However, we still get this question from those interested in migrating to Linux. One reason why this question still exists is because at one time, UNIX versions (like any other operating systems of the day) did not have a graphical user interface (GUI). By the time Macintosh and Windows appeared, UNIX systems had become somewhat obscure to the average end user.

With the increasing popularity of Linux, questions such as this are becoming less common. Still, one of the first tasks you will face is choosing the right desktop environment for your end users.

You will want to choose a desktop that your customers consider:

- Easy to use (and easy to learn)
- Easy to customize
- Easy to upgrade

Of course “easy” is a relative term, but as a consultant, you will have to quickly show the relative benefits and drawbacks of the most common desktop environments. Remember, with any UNIX-based operating system, there is more than one way to do anything. Your burden will be to provide clients with a manageable set of choices. You must also be prepared to justify the environments that you recommend.

The most common desktop environments are Gnome (www.gnome.org) and KDE (www.kde.org). Additional environments exist, including the Common Desktop Environment (CDE) and Xfce. Following is a discussion of each of these environments. We have one caveat for you, however: Please do not think that we are partial to any one of these desktops over the other. We prefer an environment provided by the Blackbox window manager, mostly because the interface is cleaner and does not imitate the Windows desktop. Making the “right” choice of a desktop environment depends on various factors, most of which can only be described by your client.

When recommending a desktop, avoid becoming a victim of the “Gnome versus KDE” wars. Simply take the time to show end users various desktop environments. Give them a chance to take a test drive, as it were. Only then will clients be able to make a relatively informed choice. So, as you read on, consider the relative strengths and weaknesses of each environment.

Gnome

The Gnome desktop was developed by The GNU Project (www.gnu.org), which is responsible for developing a wide array of software for various platforms, including Windows, Linux, and Macintosh. Figure 11.1 shows the Gnome desktop on a Red Hat Linux system.

Figure 11.1 The Gnome Desktop Environment



Understand, however, that the desktop shown in Figure 11.1 represents default settings. It is possible to customize your desktop so that it appears significantly different. When recommending Gnome, emphasize the following:

- It is associated with The GNU Project (www.gnu.org). Consequently, Gnome is licensed under GNU's General Public License (GPL), which ensures that the code is created by open-source, freely available technology.
- Many applications are written to Gnome, or use Gnome libraries. Gnome prides itself in being the desktop environment that welcomes diverse development environments, including C, C++, Tcl/Tk, and Python. Gnome's use of the GPL is also a contributing factor to the sheer diversity of applications available to Gnome.
- The code is reviewed by the same team that brings projects such as GnuPG (essentially open-source PGP), and many other applications.

- Gnome desktop developers have taken special steps to make sure that Gnome is accessible by the disabled.
- If clients want to use applications such as Galeon, Evolution, and GnomeMeeting, you might want to recommend Gnome as the default desktop.
- Gnome is often considered cleaner, because by default, it provides fewer options.

The Gnome desktop often does not appear as tightly integrated as KDE's desktop. Moreover, Gnome applications have traditionally been produced at a slower rate than KDE applications. However, many vendors have adopted Gnome because of its association with GNU, which means that the software is more likely to remain open source. To learn more about Gnome, go to www.gnome.org.

KDE

For many, KDE appears to be the most similar to the Windows environment. Figure 11.2 shows the KDE environment in a Red Hat Linux system.

Figure 11.2 Using KDE



Compare Figure 11.2 with Figure 11.1. As you can see, Red Hat took special pains to make both interfaces look remarkably similar. This does not have to be the case. When recommending KDE, consider the following benefits:

- Applications are tightly integrated. You can tell that the people who developed KDE set out from the start to design a desktop with a logical flow and a coherent organization. End users often feel that a KDE desktop provides access to more applications more quickly.
- The KDE desktop provides well-written applications that allow you to configure networking easily.
- If clients like applications such as KMail and Konqueror, you might want to make KDE the default desktop.

One of the drawbacks of the KDE environment is that it tends not to invite diverse development, as does Gnome. Therefore, you may not find as many KDE-compatible applications as in Gnome. KDE is based on the Qt GUI toolkit, and as a result, was not based on GNU's GPL. This is no longer the case. Nevertheless, this history caused many developers to adopt Gnome, at least throughout the 1990s. In addition, there is a general perception that KDE runs slower than Gnome. Personally, we have found that both KDE and Gnome run slowly, compared to more Spartan environments, such as Blackbox, discussed later in this chapter.

Notes from the Underground...

Avoiding Controversy

In some ways, we wish we hadn't discussed Gnome and KDE in terms of "benefits and drawbacks." Discussing KDE and Gnome tends to make people get quite passionate very quickly. You will have to determine the best environment for your own situation. As you present choices, make sure that you are ready to justify your recommendation based on solid business reasons, and not necessarily your own personal preferences. If you decide on a desktop environment solely based on your personal preferences, be candid about this fact. Do not try to cook up a reason that appears legitimate. Simply state your personal preferences, and then be flexible with the client.

Common Features

Both Gnome and KDE have the following priorities:

- Ease of use and customization
- Support for multiple languages

Both Gnome and KDE support Windows-like menus. Experienced Windows users should feel comfortable quite quickly, as long as they know where to access the applications they need. Both KDE and Gnome include their own versions of the “Start” application dialog box, which allows end users to launch applications that do not reside on the menu. Again, the best strategy is to give end users the chance to experience both environments.

Because both KDE and Gnome pride themselves in providing a full GUI environment, they can run quite slowly. Many people seem to have access to powerful, modern computers, so this is often not an issue.

Install Both, Make One the Default

If disk space permits, we install both Gnome and KDE. We then allow clients to choose the desktop they want to use by default. Consequently, end users can have access to both KDE and Gnome applications in either environment. Not every Gnome application is compatible in KDE, and vice versa. However, compatibility problems are increasingly rare.

We then educate end users about additional desktop environments and how to choose them. This way, we empower our clients with the ability to choose and customize their desktop environment.

Alternative Window Managers

The Xfce desktop environment was designed to run on any UNIX system, including Linux. It is also designed to be compatible with both Gnome and KDE. One of the features special to Xfce is that it supports “drag and drop” file management more completely than its competitors do. For more information on Xfce, go to www.xfce.org.

Another alternative desktop environment is the Common Desktop Environment (CDE), which was developed by a team comprised of employees from HP, Novell, Sun, and IBM. Sun Solaris systems have traditionally shipped with CDE. It is not a common window manager. For more information about CDE, go to www.sun.com/software/solaris/cde/.

The X Window System and Window Managers

The X Window system was designed to provide a standards-based GUI environment. This means that a developer who wants to create an X Window server simply needs to read common standards. He or she can then create applications that conform to those standards.

The X environment was designed from the beginning to be network compatible, meaning that it is possible to run an X Window session over the network. Consequently, using the X Window environment, you can connect to a remote system's X Window server to control it, as if you were sitting directly in front of the remote system.

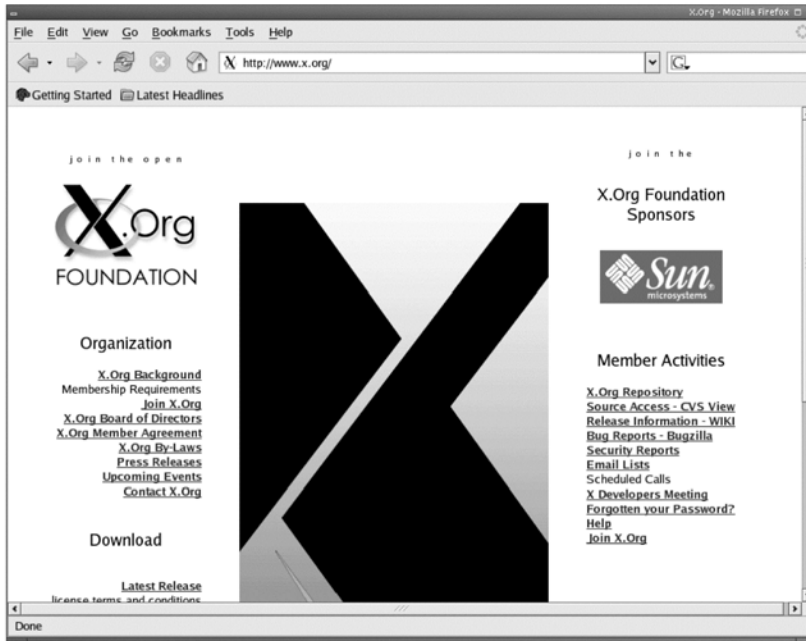
An X Window server is responsible for making sure that the GUI environment is available. This environment is most often made available to the local system, but can also be made available to remote systems. Thus, whenever you log on to the Gnome or KDE environment, you are running an X Window session. The Gnome or KDE environment is simply a client to the local system.

Two primary implementations of the X Window environment exist:

- **X.org** The X Window server used by the majority of Linux distributions, because it conforms to the GPL.
- **XFree86** Until roughly 2002, the default X Window server software for most platforms. However, XFree86 adopted a new license, dropping the GPL. As a result, many vendors and developers began supporting the X.org server.

Figure 11.3 shows the X.org Web site.

Figure 11.3 The X.org Web Site



X Window Servers versus Window Managers

A window manager mediates between the X server and the desktop environment. It is responsible for managing window toolbars and menus. It is also responsible for determining the position of applications as they are launched. Commonly used window managers include:

- **Metacity** The default window manager for Gnome desktops, after version 8.0.
- **Sawfish** The default window manager for Gnome versions 8.0 and older.
- **KWin** The default window manager for KDE.
- **Tab Window Manager (TWB)** An older window manager designed to provide only the necessary elements for a desktop. Often used during remote X sessions to ensure maximum compatibility with systems that may not have more ambitious window managers installed.
- **Enlightenment** At one time, Enlightenment was meant to be an upgrade to FVWM. For some time, however, it has been an independent

project. You can learn more about Enlightenment at www.enlightenment.org.

- **FVWM** The latest version of FVWM is FVWM2, available at www.fvwm.org.
- **AfterSTEP** You can learn more about AfterSTEP at www.afterstep.org.
- **WindowMaker** You can learn more about WindowMaker at www.windowmaker.org.
- **Blackbox** Some support for KDE, but does not officially support Gnome. You can obtain Blackbox at <http://blackboxwm.sourceforge.net>.

At least a dozen window managers exist. As you recommend a desktop environment, your goal will be to choose a window manager that makes sense for your client. If clients expect a full KDE environment that most closely imitates Windows, then you will want to use KWin. If end users want a simpler desktop, they could use WindowMaker or Blackbox. If you want a desktop that appears exactly like a Macintosh system, then choose Metacity. For more information about window managers, go to www.xwinman.org.

Tools & Traps...

Desktop Environment, X Window Server, Window Manager . . . What's the Difference?

As you consult with clients, they may not understand the difference between desktop environments, X Window servers, and window managers. Following is a brief discussion of each.

A desktop environment such as Gnome is not the same thing as a window manager. A desktop environment includes many features, such as configuration applications (for example, *yast/yast2* for SuSE Linux, or *dracnf* for Mandrake Linux) and default applications (for example, word processors, FTP applications, and calculators). A desktop environment includes a window manager. Without the desktop environment, you

Continued

would have a “bare bones” graphical environment that would alienate most users accustomed to Windows.

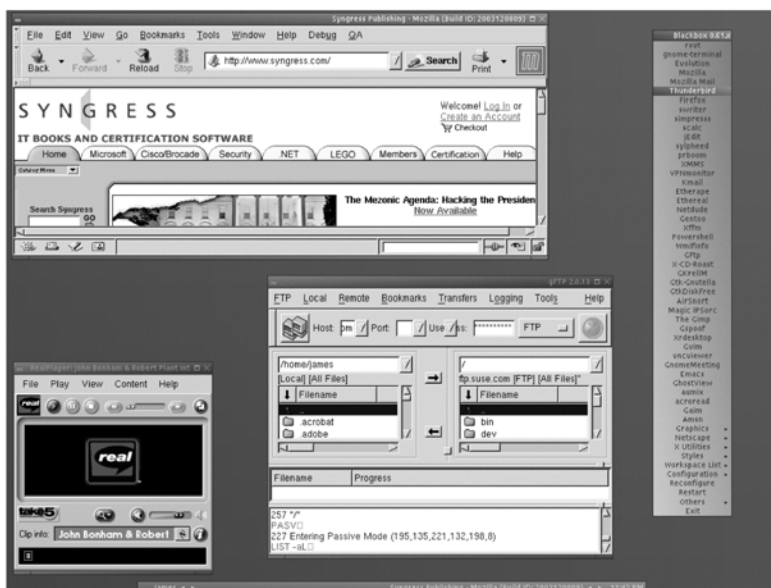
An X Window server acts as the foundation of a Linux GUI. It is responsible for providing the fonts, and the networking capability. Without the X Window server, you would not be able to have a GUI.

A window manager is a client to the X Window server (for example, one from X.org, or from the XFree86 organization). It works behind the scenes, and is responsible for the look and feel of desktop windows, including the appearance of toolbars and menus. A window manager controls how menus appear on your desktop, too. If you can access a Linux system, begin an X Window session and then launch any application. Look at the title bar to the application. Notice how the application is launched into a certain portion of the screen (for example, in the center, or to the left). Use your mouse’s right and left buttons. These elements are all controlled by your window manager. Without a window manager, the content served up by the X Window server would be incoherent, and would not have a common theme.

Window Managers as Alternative Desktop Environments

You are not limited to Gnome, KDE, CDE, and XFCE. Alternatives to the Blackbox window manager are shown in Figure 11.4. Blackbox is quite different from both Gnome and KDE. For example, it does not have Windows-like menus or taskbars. In addition, Blackbox is a window manager, and not simply a desktop environment.

Figure 11.4 The Blackbox Environment



You simply right-click on the desktop to make the menu appear. You can then select the applications you want to run. One of the benefits of an environment like Blackbox is that it is less resource intensive, and thus loads faster. We prefer speed in any case, mostly because we do not have the money to purchase a new system each time a Gnome or KDE developer introduces a new process-hungry GUI feature.

Notes from the Underground...

What Does the Customer Want?

Your job as a consultant migrating customers from Windows to Linux desktops is to:

1. **Identify the customer's needs.** Determine the services that the customer wants. Create a detailed list of the customer's needs. Determine right away if an open-source alternative exists. If Linux is not part of that solution, do not try to force-

Continued

fit Linux into a company. If you do, you won't be asked back again, and you will be saddled with a disgruntled customer.

2. **Identify solutions.** Your job is to understand the open-source choices that exist. Learn about the latest solutions. Frequently visit sites such as Freshmeat (www.freshmeat.net), SourceForge (www.sourceforge.net), and even Slashdot (www.slashdot.org) to remain informed concerning the latest software developments.
3. **Fulfill the needs of the customer.** Using Linux-based applications, create feasible, workable solutions that allow the end user to access the desired services and obtain the desired information with minimal retraining. You will first need to run an extensive test deployment to ensure that your solution truly meets the customer's need. Another step includes conducting a final acceptance test. Your customers will want to have a "grace period" so that they can determine if your solution is working properly. Finally, you will need to properly train end users so they understand the solution.

Even the most experienced consultants have failed to please their customer at one point or another. You will find that whenever a consultant runs into trouble, he or she has failed to follow the preceding three steps.

E-Mail and Personal Information Management Clients

E-mail and PIM have become closely related, because most people communicate their availability through e-mail these days. This section discusses e-mail and PIM software that will help users remain organized, even without Outlook.

It has been our experience that most end users think Outlook *is* e-mail. Many of our clients don't realize that they are simply using an application to send and get their e-mail. As you work with clients, take the time to determine if they understand the difference. It might be tempting to make fun of people who cannot differentiate an e-mail application from an e-mail itself. The problem is, many of these end users we so often complain about run companies, hire (and fire) employees, and make purchasing decisions. It is your job to inform them that they can use e-mail, even if they no longer use Outlook or Outlook Express.

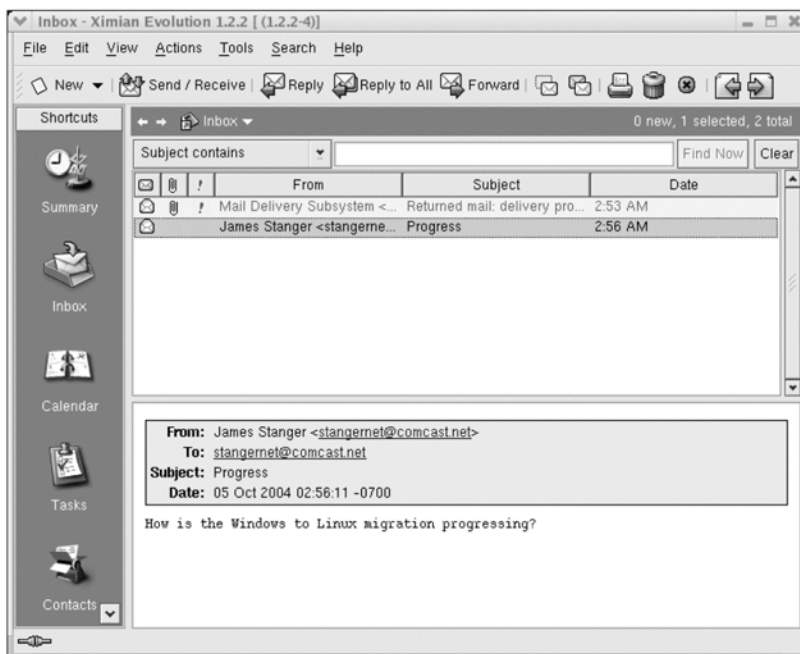
Therefore, for those end users who aren't sure that e-mail exists outside of Outlook or Outlook Express, try giving them the following section to read, because there is certainly no shortage of e-mail applications in the Linux space. Commonly used e-mail applications include:

- Evolution
- KDE Suite/KMail
- Mozilla mail/Thunderbird
- Aethera
- Sylpheed

Evolution

Evolution is Gnome's default mail and PIM client. KMail and Evolution will run in any window manager you decide to use. They will also run inside the KDE, Gnome or Blackbox environments. Figure 11.5 shows the Evolution e-mail interface. From here, you can send and receive e-mail.

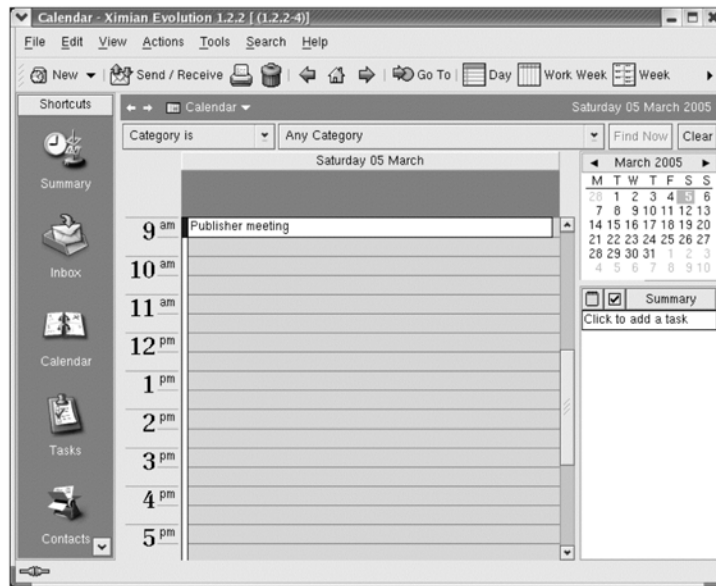
Figure 11.5 Evolution and the E-Mail Interface



Evolution stores its files in the mbox format. If your username is james, look for e-mail in the directory `/home/james/evolution/local`, which contains directories for all of your mail folders. Inside each folder you will find a file called `mbox`, which is your mail in mbox format.

Evolution also has PIM features, including the calendaring feature shown in Figure 11.6.

Figure 11.6 Evolution Showing the Calendaring Interface



Benefits of using Evolution include:

- It is developed by Novell, a company with a history of developing a solid customer base.
- It will run on any common window manager (for example, KDE or Gnome).
- It is designed to work with common groupware servers, such as Microsoft Exchange.

Evolution, Microsoft Exchange, Novell GroupWise, and OpenExchange

Evolution is unique in that it works well with servers created by other vendors. For example, Evolution's Connector for Microsoft Exchange allows you to take advantage of all features provided by Exchange. Similarly, plug-ins for Evolution allow it to act as a client for Novell GroupWise, and Novell's OpenExchange servers. You can learn more about Evolution at www.novell.com/products/evolution.

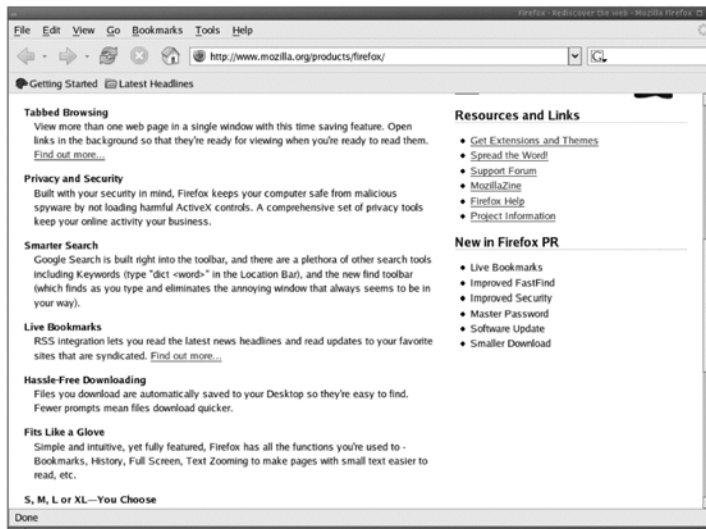
KDE Suite/KMail

KDE's default mail client is called KMail. It can either be run by itself or built into Kontact, which makes it look more like Outlook. In KMail and most other mail clients, all your mail will end up in your home directory in a folder called Mail unless you are running IMAP. Inside your `/home/user_name/Mail` folder are all of your mail files such as inbox, trash, sent, drafts, and so forth. Copy the files over and make sure you set permissions correctly so the user has sole read and write permissions on them. Your Mail folder should look something like this:

```
ls -lh /home/james/Mail
total 11M
-rw----- 1 james james 0 Aug 20 19:51 drafts
-rw----- 1 james james 11M Aug 20 19:51 inbox
-rw----- 1 james james 0 Aug 20 19:51 outbox
-rw----- 1 james james 26K Aug 13 19:04 sent-mail
-rw----- 1 james james 0 May 17 18:32 trash
```

Figure 11.7 shows the KMail application.

Figure 11.7 KMail



Kontakt

Kontakt is essentially KMail on steroids. It allows you to connect to the following groupware servers:

- **Microsoft Exchange** Currently, Kontakt supports Microsoft Exchange 2000 only. For more information, go to www.microsoft.com.
- **Novell GroupWise** Currently, Kontakt supports version 6.5. For more information, go to www.novell.com.
- **eGroupWare** A PHP-based groupware application designed by and for the open-source community. Runs on Linux servers. For more information, go to www.egroupware.org.
- **The Kolab project** A groupware server first established by the German government. For more information, go to www.bsi.bund.de.

Thus, KMail is a competitor (or should we say, kompetitor?) to Evolution. You can learn more about Kontakt at www.kontakt.org.

Aethera

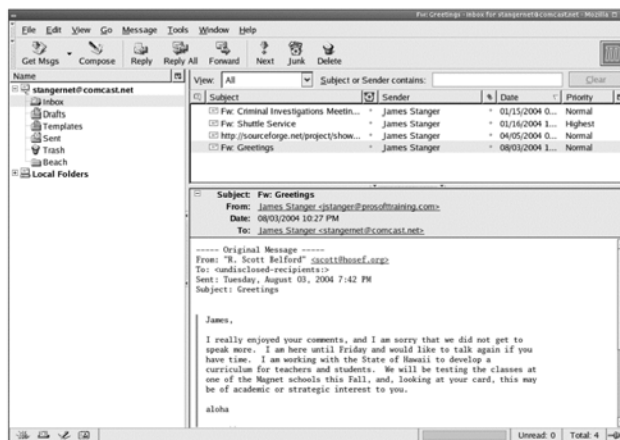
Like Evolution, Aethera is e-mail application with bundled PIM software, and was written to the GPL. However, Aethera is designed to support only the Kolab groupware server as of this writing. Figure 11.8 shows Aethera's calendaring feature.

Figure 11.8 Aethera's Calendaring Feature

Aethera is a GPL application, and is considered quite reliable. However, its limited groupware support may be a problem for those companies that do not want to migrate to a Kolab server. You can learn more about Aethera at www.thekompany.com/projects/aethera/index.php3.

Mozilla Mail/Thunderbird

Mozilla Mail, shown in Figure 11.9, is bundled in with the Mozilla Web browser and Composer, a GUI HTML editor. Mozilla Mail is a capable e-mail client, and supports SMTP, POP3, and IMAP.

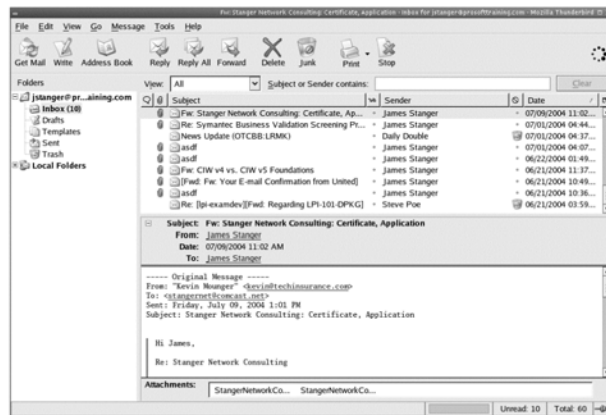
Figure 11.9 Mozilla Mail

Mozilla Mail is extremely common, and it is a stable product. However, it is not a groupware client as of this writing. Because it is bundled with an HTML editor and a browser, it is a perfect recommendation for companies that do not use groupware. Many companies will find that their end users can take advantage of the HTML editor, and the Web browser. You can learn more about Mozilla Mail at www.mozilla.org. In addition, read the subsequent section, “Migrating E-Mail.”

Thunderbird

Even though Thunderbird is also created by the developers at Mozilla.org, it uses different code than Mozilla Mail does. Thus, Thunderbird deserves to be treated as a different application. Figure 11.10 shows the Thunderbird application.

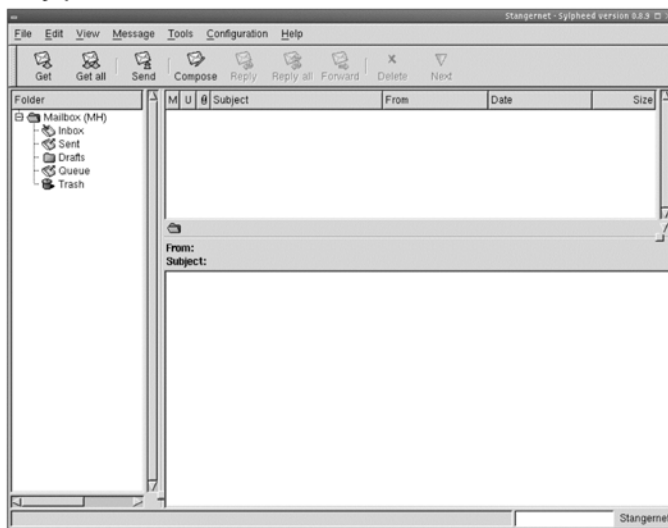
Figure 11.10 Thunderbird Application



Like Mozilla Mail, Thunderbird does not support groupware servers as of this writing. However, it loads fast, and has a small footprint, making it ideal for those who need a stand-alone e-mail application. You can learn more about Thunderbird at www.mozilla.org.

Sylpheed

Sylpheed, shown in Figure 11.11, is one of many e-mail applications that exist in the market. It does not have groupware or scheduling features. However, it does one thing quite well: it supports PGP and GPG. Although many clients say that they support PGP and GPG, few work as well as Sylpheed.

Figure 11.11 Sylpheed

Thus, if your clients need to use PGP or GPG, consider recommending Sylpheed. To learn more about Sylpheed, go to <http://sylpheed.good-day.net>. Sylpheed's creators have focused on making sure that it supports IPv6, which is the next version of IP, designed to improve security.

Essential Information

No matter what client you plan to use, you will need the following information:

- The SMTP server name or IP address
- The POP3 or IMAP server name or IP address
- User authentication information (for example, username and password)

Take the time to write down this information so it is handy. As you conduct a migration, you will not want to repeat this information constantly to those who are helping you.

Recommending E-Mail and PIM Software

Don't think that you have to recommend just one e-mail or PIM application. In many cases, you may have to install multiple applications to get what you want. For example, you may need to use Evolution to allow clients to connect to the company Exchange server and use their calendars, but use KMail in order for

clients to use their Internet e-mail. It is more likely, however, that your clients will want to standardize to one application.

You will likely find that your clients who preferred Outlook will prefer Evolution. Clients who already use the Windows Mozilla/Netscape/Thunderbird variants will find it very easy to transition over to their Linux counterparts. Some of your clients may still be using Eudora. These customers will likely choose KMail, as it has a relatively similar look and feel. You can be confident in recommending these clients; they are all quite stable and feature rich.

Migrating Mail

If your customers are using something other than Outlook for their mail clients, you may not have to do a conversion, as your mail is most likely already in a mbox format. However, if you are using Outlook, you have to convert the format in which your mail is stored. There are five ways to do this. We will start with the easiest, and move to the “if all else fails” way last. Depending on the version of client that is in use, you might have to do an upgrade before you can move the mail. This part of the migration can take a long time per person if you need to migrate thousands of e-mails per client.

One of the best ways to ensure that you have time to do a proper migration is to plan a staged migration, which involves using multiple applications to convert your e-mail. You may find that you cannot export mail directly from a Windows e-mail application, and then import it into a Linux application. You will find that you must first export e-mail into an intermediary e-mail client that can then export your e-mail into the format required by the e-mail application you want to use.

Now, let's look at the steps necessary when migrating from Outlook and Outlook Express.

Migrating from Outlook or Outlook Express

The first step is to do a backup in case you have any problems. In Outlook, you want to export your e-mail messages to a single .pst file. This is done by clicking on **File | Import and Export** to bring up the Import and Export Wizard dialog box. You then select **Export to a file**, as shown in Figure 11.12.

Figure 11.12 The Microsoft Outlook Import and Export Wizard Window

After clicking **Next**, choose **Personal Folder File (.pst)**. Then, select the top of the tree by clicking on **Personal Folders** and make sure to select the check box **Include all subfolders**. Make sure that you remember where you save the backup.pst file. As you save the file, do not export your backup using any encryption, compression, or password protection. If you do, the import process will fail. You can now import the file into Mozilla.

Importing Outlook Mail into Mozilla

Now, install Mozilla on the Windows desktop system. Make sure you select Mozilla and not Firefox or Thunderbird for this step, even if you do not plan to run Mozilla at the end of the process.

First, install Mozilla and select **Complete** when asked which components you want to install. You do not need to tell the system to make Mozilla the default browser or e-mail client. Once it's installed, start Mozilla and click on **Windows>Mail & Newsgroups**. You do not need to fill in the settings in the account wizard that will pop up. You can import these into Mozilla from Outlook if you're so inclined; it will appear as a second account.

Once you have the account set up you need to import the old mail out of Outlook into Mozilla. Now, do the same for the settings and the address book. If you prefer, you can just export the address book in .csv format from Outlook or Outlook Express and import it into Mozilla the same way as the mail.

To do so, click on **Tools>Import** on the Import screen, select **Mail**, and select the mail client your importing from; it will do the rest. Once you have completed these steps, you will have your mail converted from a .pst format to a standard mbox mail format. In our case, the imported mail folders end up in C:\Documents and Settings\james\Application Data\Mozilla\Profiles\default\033c70c1.slt\Mail\Local Folders\Outlook Mail.sbd\Inbox.sbd.

Inside this folder are all the folders that existed in the Outlook client. You will notice you have two files for every folder: the mail file itself, and an index of what is in the file. If you try to open the file without the .msf extension in something such as Notepad, you will see it's a standard mbox mailfile. Some people have reported that Mozilla does not do the conversion.

Now you can copy all of these files over to the new system in whatever fashion you prefer—burn them to a CD, FTP them to the server, or use winscp to copy them over to the new system.

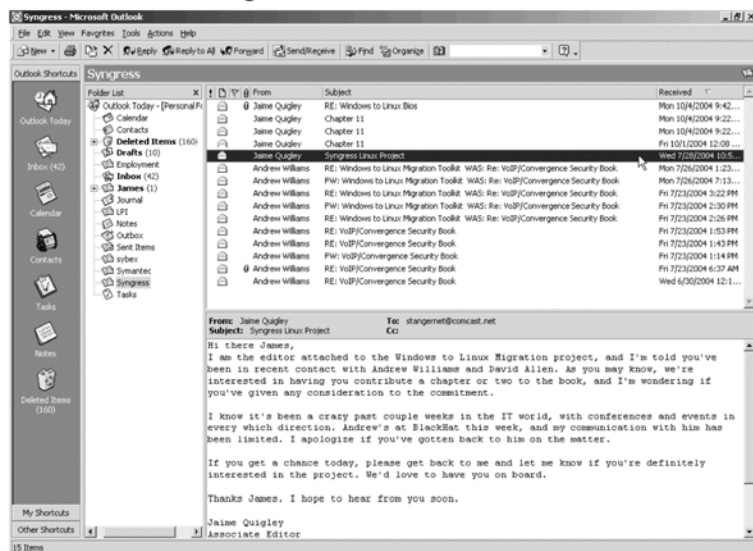
LibPST

LibPST is a Linux application that converts PST files into Mozilla-compatible mbox files. Therefore, once you have generated a PST file, you simply install and use LibPST on your Linux system to prepare the contents of your PST file for use with Mozilla. You can obtain LibPST from <http://sourceforge.net/projects/ol2mbox>. LibPST is ideal if you have a particularly large PST file that you need to convert. In many cases, Mozilla will fail to process it using its own conversion utility.

Importing Outlook Mail into Evolution

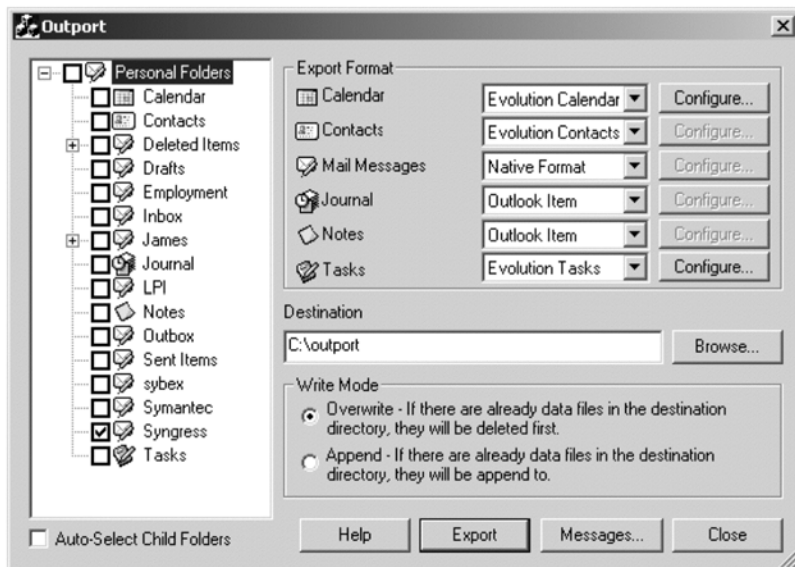
One way you can import Outlook Mail into Evolution is to use an application called Outport. You can download Outport from <http://outport.sourceforge.net>. For example, suppose a client has a group of e-mail messages named Syngress in Outlook Express, similar to that shown in Figure 11.13.

Figure 11.13 E-Mail Messages in Outlook



All you have to do is download Outport, and then double-click on the outport.exe application. The Outport main interface will appear, as shown in Figure 11.14.

Figure 11.14 Using Outport to Export Messages from Outlook

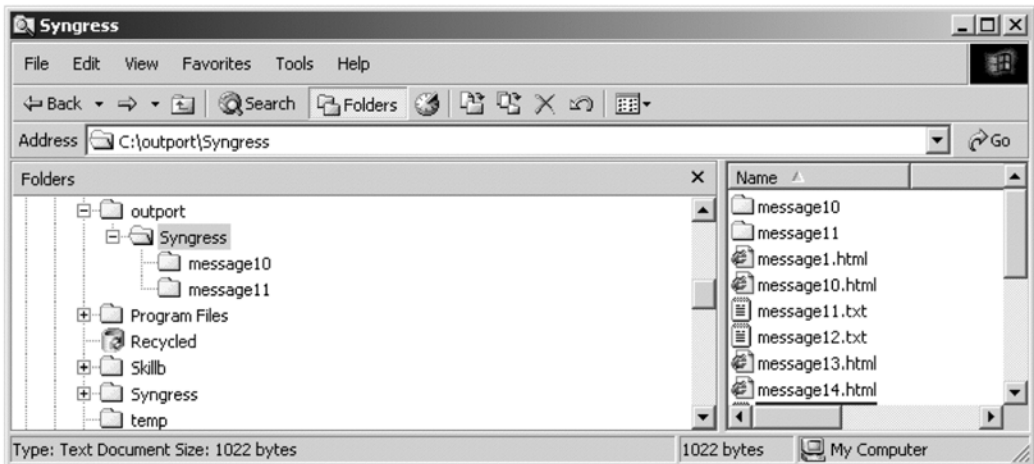


When the Outport interface appears, you can choose various Outlook elements to export, including:

- Outlook Calendar entries
- Contacts
- Mail messages
- Journal entries
- Notes
- Individual tasks

Before clicking the **Export** button, you will need to specify a destination directory. In the preceding example, the `C:\outport/` directory is used. After clicking **Export**, Outport will convert and export the files into the Outport directory, where you can view them in a file manager, such as Windows Explorer, shown in Figure 11.15, or Gentoo, a file manager for Linux (www.obsession.se/gentoo).

Figure 11.15 Viewing Exported Messages in Windows Explorer



Once you have exported these files, you can then import them into Evolution, or into another application so that you can more easily convert them into files that Evolution understands.

Document Standards

As you migrate e-mail and PIM software settings from Windows systems to Linux, you will have to become familiar with the following standards:

- **The Internet Calendaring and Scheduling Core Object Specification (iCalendar) standard** Known as Ical, it is defined in RFC 2445. Apple machines were among the first to adopt this standard. Used for personal calendars.
- **The Vcalendar standard (Vcal)** Used to for making appointments.
- **Virtual Card standard (Vcard)** An electronic business card format meant to provide a uniform, textual representation of an event. It is a cross-platform way to represent a calendar event. Outlook can export into these formats (see Chapter 8, “Groupware and Calendaring Services”). This standard is defined in RFC 2426, and is used often used in PIM software.

The Hard Way

The least favorable migration option is to simply forward e-mail from a Windows client to the new Linux system. If you have to resort to this method, perhaps the easiest way is to forward an entire mail folder. Still, forwarding a large number of e-mail messages as one e-mail will probably guarantee that end users will never find the e-mails again, unless they work hard to sift through (perhaps) dozens of attachments buried in a folder. Therefore, use your best judgment when forwarding e-mail.

Web Browsers

Web browsers are not just used to “surf the Web”; they can be used to launch embedded applications, check e-mail, and view groupware calendars. As a result, Web browsers are increasingly sophisticated, and must support various authentication and encryption schemes. In this section, we discuss how to choose the appropriate browser(s) for your client.

You will find that end users will expect their Linux systems to offer one (and only one) browser. With Windows, everyone tends to use Internet Explorer. This browser has been on every computer since Windows 98; many end users became accustomed to Internet Explorer on Windows 95 systems.

With Linux, however, there are several choices, including:

- Mozilla
- Firefox
- Galeon
- Konqueror
- Opera

It is likely that no one Linux Web browser will meet everyone's needs. You will have to become familiar with many of the browsers available. Following is a discussion of the most important GUI-based browsers.

Mozilla

Mozilla, shown in Figure 11.16, is actually a group of applications that includes the Mozilla browser, Mozilla Mail, and Composer.

Figure 11.16 Mozilla



Mozilla's advantages include:

- **Tabbed browsing** The ability to see multiple pages in one window allows you to browse more efficiently.
- **The Gecko rendering engine** An engine that renders the Web pages quickly and efficiently.
- **Speed** The browser renders pages quickly, and loads into memory quite easily.
- **Stability** The code is extensively reviewed, and can be more stable than many other vendor browsers.
- **Built-in pop-up blocking** You do not need to download third-party software to block pop-up ads, if you are using Mozilla.
- **Built-in applications** You can view newsgroups, and can use an Internet Relay Chat (IRC) client.

If your customers need a full-service package and do not want to deal with separate applications, recommend Mozilla.

Mozilla and Microsoft CHAP

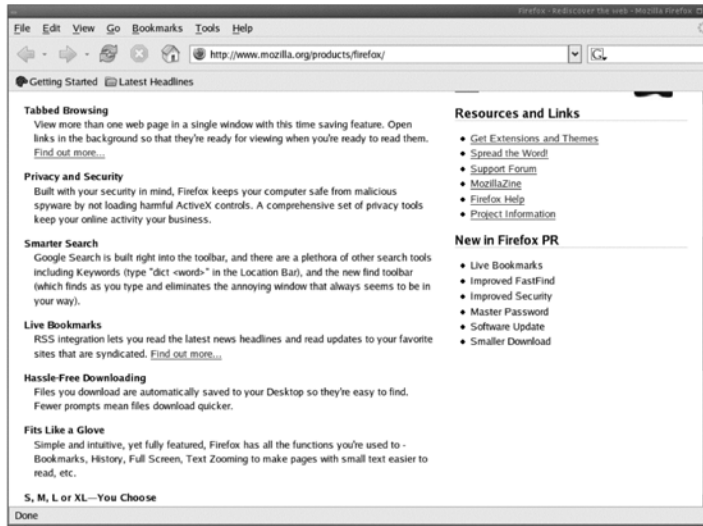
The Internet Information Services (IIS) Web server uses a special form of the Challenge Handshake Authentication Protocol (CHAP), called MS-CHAP. Microsoft designed IIS this way. The intent was that if an IIS administrator was to enable MS-CHAP, only those using Internet Explorer could authenticate securely.

However, Mozilla.org was able to implement MS-CHAP as of version 1.6. This is an important development, because it eliminates one more reason for remaining with Internet Explorer, which has experienced the most serious security problems.

Firefox

Firefox is a stand-alone browser based on the Gecko engine, just like Mozilla (see Figure 11.17).

Figure 11.17 Firefox



However, Firefox is not simply another form of Mozilla. Firefox has the following features:

- **Improved customization** From font choice to determining which buttons you will see, Firefox is designed to allow more customization.
- **Faster page rendering** Firefox has a more current version of the Gecko engine.
- **Smaller footprint** The Firefox developers have tried to keep the application's size down to roughly 4MB.

If your customers do not mind using separate applications (for example, separate Web browsers and e-mail clients), then recommend Firefox. Finally, Firefox benefits from MS-CHAP compatibility. You can learn more about Firefox at www.mozilla.org.

Galeon

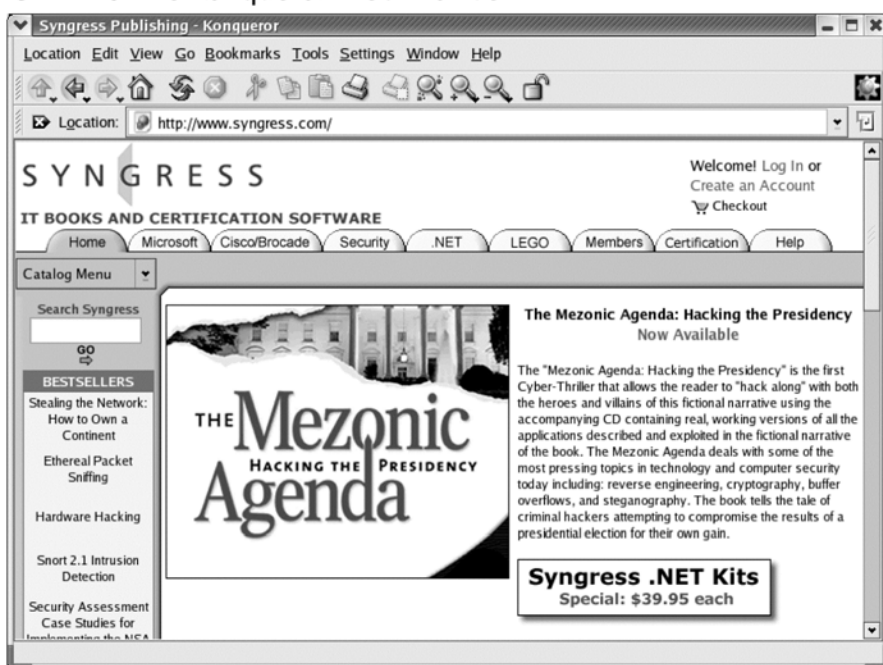
Galeon is specially designed for the Gnome desktop. However, it uses Mozilla's Gecko rendering engine. Consequently, it is a solid choice for clients who want the stability and speed of the Gecko engine, and want to take advantage of the Gnome desktop environment. As good as Mozilla and Firefox are, they are not specifically designed to run in Gnome. Therefore, Galeon may load and render

faster than any browser in an optimized Gnome environment. For more information about Galeon, go to <http://galeon.sourceforge.net>.

Konqueror

Konqueror, shown in Figure 11.18, is KDE's default browser. It uses the KHTML rendering engine. Interestingly, MacOS' new browser, Safari, uses the same rendering engine. If you are migrating end users over from Macintosh systems, Konqueror might be the best choice.

Figure 11.18 The Konqueror Web Browser



Of the browsers profiled in this chapter, Konqueror is the only one meant only for Linux/UNIX systems. It is not a cross-vendor browser.

Opera

Opera is the only fee-based browser discussed in this chapter. To some companies, paying for software actually brings a sense of security. Paying for software, many feel, results in a stronger support contract. Opera's developers argue that Opera has the following advantages:

- The fastest rendering of any Web browser
- Tabbed browsing
- The ability to focus and “zoom in” on content
- IRC compatibility

In many ways, Opera offers the same features as Mozilla.

Migrating Bookmarks

You will find that migrating end users to a Linux browser is relatively easy, because virtually every browser that runs Linux automatically imports the bookmarks exported from Internet Explorer. The exported bookmark data is easily found. However, with recent versions of Internet Explorer, this data is stored in a directory. On our system, the Internet Explorer bookmarks are found in C:\Documents and Settings\james\Favorites.

Once you have installed Mozilla, Firefox, or Opera, this data is easier to get to since it will be stored in a single file. This file can be called `bookmark.htm`, `bookmarks.htm`, `bookmark.html`, or `bookmarks.html`, depending on the version of the browser. All you have to do is copy this file to the new Linux system when you migrate the operating system. Then, simply find the feature for managing bookmarks. In Firefox, for example, you would go to **Bookmarks | Manage Bookmarks**. When the Bookmarks Manager window opens, go to **File | Import**, and then select the **From File** option button. You can then select the Internet Explorer bookmark file you exported.

Browser Plug-Ins

Windows users are likely aware of dozens of browser plug-ins. In Linux, you will find a more limited set of plug-ins, although most of the essential ones are supported, including:

- Macromedia Flash, and Shockwave/Director
- RealNetworks Realplayer
- Adobe Acrobat Reader

Following is a discussion of each plug-in technology.

Macromedia Flash and Shockwave/Director

Macromedia Flash has been available in Linux for a number of years. It is relatively easy to install. However, Flash does not automatically upgrade itself in Linux as it does in Windows browsers. Make sure your clients understand that they will have to manually update their Flash plug-ins periodically.

Installing Flash is as simple as downloading it from Macromedia's Web site at www.flash.com. Once you download the file (for example, `install_flash_player_7_linux.tar.gz`), you simply unzip and untar it and then change to the `install_flash_player_7_linux/` directory. You then run the `flash-player-installer` script (for example, `flashplayer-installer`) and follow the onscreen instructions.

As of this writing, Macromedia Shockwave/Director can only be installed in Linux by using the CrossOver Office Plugins bundle, which we discuss later in this chapter. However, this will likely change, as Macromedia has begun demonstrating true interest in Linux, now that its market share has increased. Because there is virtually no business justification for it, though, few people other than graphic artists or home users will need this functionality. In short, if you need Shockwave/Director, install the CrossOver Office Plugins bundle.

RealPlayer

RealPlayer, shown in Figure 11.19, is one of the more essential plug-ins, because it allows you to view streaming media. You can use it for both streaming audio and video.

Figure 11.19 RealPlayer



You can run RealPlayer from any of the Web browsers profiled in this chapter. For example, in Konqueror, you will be asked which application you want to use. Simply enter the name of the RealPlayer binary (for example, `realplay`), and you will be able to see or hear the media you have chosen. You can download the basic version of RealPlayer free of charge at www.realplay.com. You will have to consult with your customers concerning which version to use.

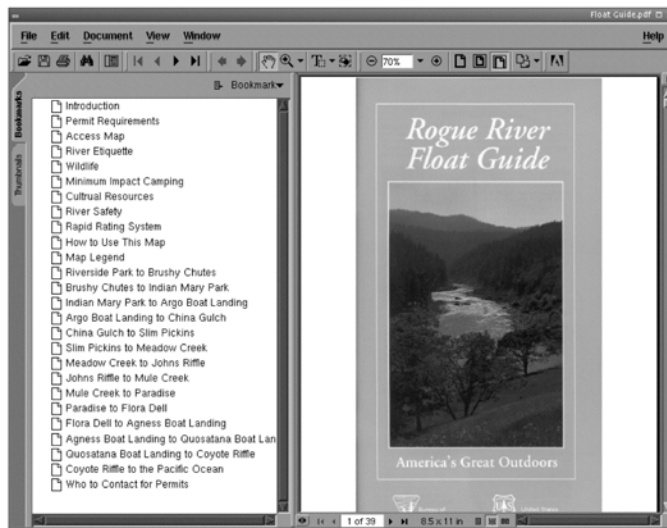
You may have to specify RealPlayer as a “helper application” in order to properly run streaming video. To do this, open the **Preferences** in any browser and access the appropriate window that allows you to define file associations. For example, in Konqueror, you would go to **Settings | Configure Konqueror**, and then select the **File Associations** screen. In Mozilla, you would go to **Edit | Preferences**, select **Helper Applications**, and then define the appropriate MIME type. Common MIME types for streaming media include:

- `application/x-pn-realaudio` (for `rm` and `ram` files)
- `audio/x-realaudio` (for `ra` files)
- `audio/x-wav` (for `wav` files)

Adobe Acrobat Reader

Another essential plug-in is Adobe Acrobat Reader, shown in Figure 11.20.

Figure 11.20 Acrobat Reader



You may have to define the MIME type for PDF on certain browsers. MIME types you can define include:

- application/pdf
- application/x-pdf
- application/acrobat
- text/pdf
- text/x-psdf

We have always found application/pdf to be sufficient. Acrobat Reader is available free of charge at www.adobe.com.

Office Application Suites

You have already learned about migrating workstations to Linux-based e-mail/PIM clients and Web browsers. You will likely find a way to make end users happy managing e-mail and using browser-based applications. However, end users do not simply use e-mail and Web browsers all day; they will need to create documents and presentations. “Where,” end users will ask, “is Microsoft Word?” They will also want to know where their Excel and PowerPoint applications are. In other words, end users will want to know if they will be able to work with their files.

End users will want to do their jobs, and will not want the operating system to “get in the way.” Managers will be concerned about lost productivity. You will, of course, have to assure these parties that even though they will be migrating from Microsoft Office, they will remain productive. You will have to show that users will:

- Still be able to work with their .doc, .rtf, .xls, and .ppt files.
- Become comfortable with new applications in an acceptable period of time.
- Be able to exchange files with others who will still be using Microsoft software.

The most common office software suites include:

- OpenOffice.org
- StarOffice

- KOffice
- Hancom Office

The following sections will discuss each suite.

OpenOffice.org

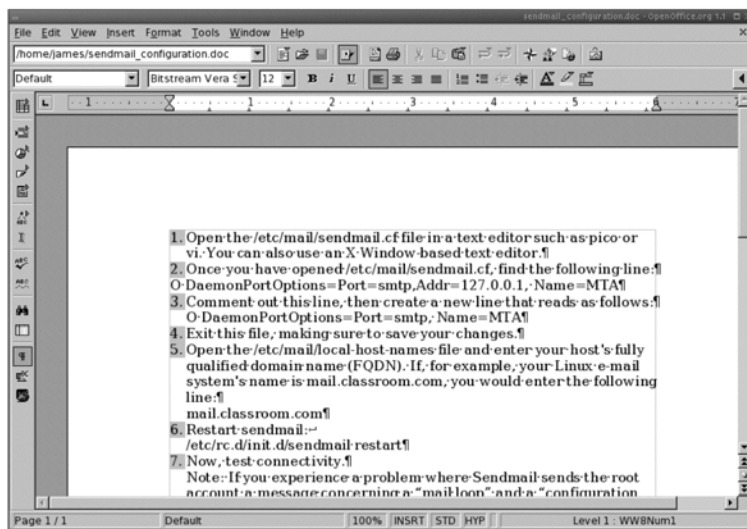
OpenOffice.org quickly became the standard for the Linux office suite when Sun introduced it in 1999. OpenOffice has several applications. The following are the most often used:

- **Star Writer (swriter)** A word processor; the equivalent of Microsoft Word.
- **Star Impress (simplpress)** Slide presentation software; the equivalent of Microsoft PowerPoint.
- **Star Calc (scalcc)** Spreadsheet software; the equivalent of Microsoft Excel.
- **Star Web (sweb)** Web page creation software; the equivalent of Microsoft FrontPage Express.

Basically, very little needs to be done to OpenOffice once you have it installed. If you share documents outside of your company, as many people do, you can make it easier on your users by telling OpenOffice to save in .doc, .xls, or .ppt format natively. This is done by clicking on **Tools>Options>Load&Save>General**, and in the standard file format section, make sure you select **Always save as Microsoft Word 97/2000/XP for Document type, text document**. Do the same for **Spreadsheet and Presentations**. Make sure you do not select templates, though, as this has to be done differently. You can learn more about OpenOffice at www.openoffice.org.

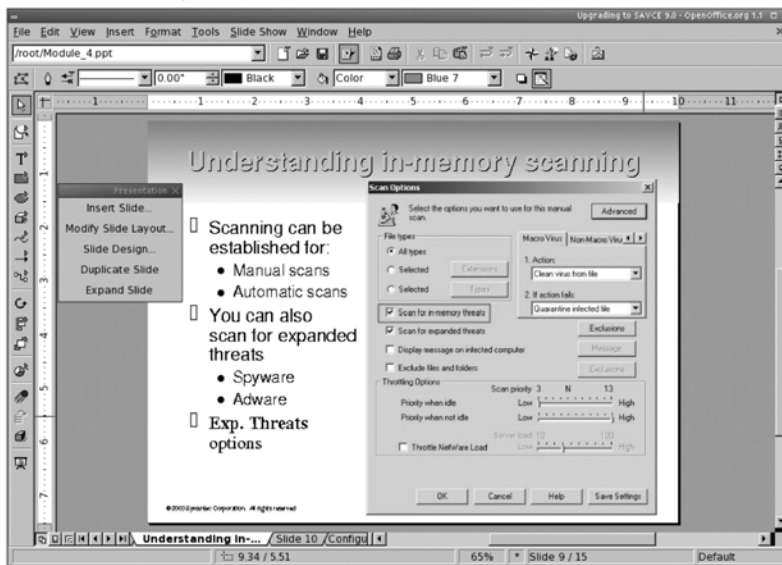
OpenOffice.org can open any document created in Microsoft Office 2000. For example, the document shown in Figure 11.21 was created in Microsoft Office 2000.

Figure 11.21 Star Writer



The document shown in Figure 11.21 was also modified by Office 2003. Figure 11.22 shows the Star Impress program. You can see that the interface is quite similar to PowerPoint.

Figure 11.22 Star Impress



The document shown in Figure 11.22 was created in PowerPoint, and then opened in Star Impress. In fact, the majority of the slides created for this presentation were created in Star Impress. The presentation was then sent to several individuals who used nothing but Microsoft PowerPoint. They were completely oblivious that the slides were created on a Linux system.

Figure 11.23 shows the Star Calc application opening a simple spreadsheet.

Figure 11.23 Star Calc

	A	B	C	D	E	F	G	H
1	Mitel 3300 Exam Cost Estimate and Timeline							
2								
3	Actions to Create "CTP - UK version"			Cost Estimate Calculation	Projected Costs	Start Date	End Date	
4	1	SME to write 240 questions based on the Mitel 3300 Installation and Maintenance course. SME should be able to develop questions at a rate of 6 a day. This will take 40	Developer will be paid \$47.52 an hour. We assume 40 8-hour days (320 hours), and multiply this figure by \$47.52.	\$30,000	08/1/03	10/3/03		
5	5	Mitel SME to review questions. This should take 4 days.	At Mitel's expense.	N/A	10/06/03	10/09/03		
6	2	ProsoftTraining Editor to edit final questions. This should take two weeks and a day.	Editor will be paid \$33.12 an hour. We assume 11 8-hour days (88 hours), and multiply this figure by \$33.12.	\$2,915	10/10/03	10/24/03		
7	3	ProsoftTraining Publisher to format questions into a format of Mitel's choosing. This should take four days. If any special software or formatting is required, we will need paid software and may require grade training from a Mitel representative. Extensive formatting may extend this proposed due date, or necessitate a multimedia developer, who operates at a higher rate. Timeline includes additional	Publisher will be paid \$24.48 an hour. We assume 4 8-hour days (32 hours), and multiply this figure by \$24.48.	\$263	10/27/03	10/31/03		
8	4	ProsoftTraining Project Manager to coordinate activities and prepare final deliverable. Approximately 4 days.	Project manager will be paid \$20.80. We assume 4 8-hour days (32 hours), and multiply this figure by \$20.80.	N/A	N/A	N/A		
9			Projected Costs					
10			48% markup for new development	\$7,930				
11			Total Billing to Mitel	\$27,756				
12								

The spreadsheet in Figure 11.23 is relatively simple. However, most individuals will do little more with a spreadsheet than create rows and columns, and then calculate summaries. Notice also that this particular spreadsheet supports tabbed sheets.

Limitations: Macros and PDF Files

With very few exceptions, PDF files are all easily read, created, or modified by OpenOffice. The exception to this rule is if you're using macros. While these won't run natively on OpenOffice or any of the other alternatives, there is a way to rewrite your macros to work with OpenOffice, as it contains its own very powerful Macro Writer.

OpenOffice.org has the capability to create PDF files “on the fly.” This feature is quite impressive, but is still not completely perfected. It is anticipated that future versions will offer production-quality PDF generation.

We have found that more complex templates do not work well in OpenOffice. This suggests that more complex work, such as that done by desktop publishers, may be too ambitious for OpenOffice.org. However, OpenOffice.org can handle the vast majority of your word processing, slide presentation creation, and spreadsheet development demands.

Nevertheless, the average end user will not encounter any difficulty using OpenOffice. Only those end users who engage in truly advanced word processing and spreadsheet creation will encounter problems. Microsoft Office 2003 has introduced features not supported by OpenOffice.org, as well as Office 2000 and Office XP. For example, Microsoft Office 2003 allows you to create editing restrictions on certain portions of a file. OpenOffice.org does not support this feature. It is important to understand, however, that many people have not upgraded their Office 2000 software to Office 2003, and that they, too, do not take advantage of these features.

One other weakness of OpenOffice.org is macro support. You cannot simply import all of your macros into StarOffice. At its best, recreating macros is never particularly fun. At its worst, it is an enormous time burden. If you require extensive macro support, we suggest that you read about CrossOver Office.

So, as you introduce office suites such as OpenOffice.org, do not state or even imply that these suites will replace Microsoft Office painlessly. In some cases, you and your customers will be disappointed. Make sure that you determine exactly what your customer needs, and then thoroughly test your solution before recommending it.

Future Plans

It is likely that OpenOffice.org will be able to generate Shockwave Flash (SWF) files on the fly. Thus far, no office suite has this capability. OpenOffice.org already uses XML as the basis of its files. Consequently, this suite is capable of handling complex tasks. It is simply going to take time for OpenOffice.org to equal products such as Microsoft Office.

StarOffice

StarOffice is essentially a stable version of OpenOffice.org with the promise of customer support. Perhaps an analogy will help: StarOffice is to OpenOffice.org

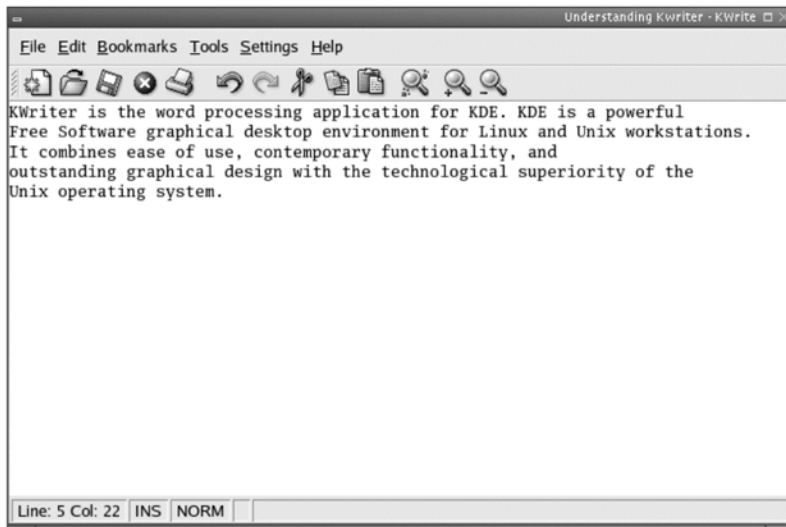
much like Netscape Navigator is to Mozilla. Much like Netscape takes a stable version of Mozilla and then sells it, Sun takes OpenOffice.org and sells it as StarOffice.

Corporations are becoming increasingly fond of StarOffice, because they can purchase customer support in case of a problem or bug. For a nominal fee, you receive customer support, Internet-based file storage, and increased macro support. You can learn more about StarOffice at www.staroffice.com.

KOffice

A project of the KDE team, KOffice is a fully functional replacement for Microsoft Office. Figure 11.24 shows the KWrite application, the default KDE text editor. While more powerful than a simple text editor such as Wordpad or Notepad, it is nevertheless not as powerful as Star Writer.

Figure 11.24 KWriter



Hancom Office

Hancom Office is sold by a Korean company that has the following goals:

- Create a user-friendly suite that is compatible with Microsoft Office.
- Support a large number of languages. Extensive Unicode support means that Hancom office is also ideal if you need to support offices that pro-

cess documents in Chinese (both simplified and traditional), Korean, and Arabic.

You can learn more about Hancom Office by going to www.hancom.com. One of the ways in which Hancom Office tries to ensure compatibility with Microsoft Office documents is an automatic update feature. This feature allows you to quickly obtain the latest filters and rendering engines.

One of the limitations of Hancom Office is that it does not support macros written in Visual Basic. Hancom Office does not improve upon OpenOffice.org or StarOffice in this regard.

Running Windows Applications on Linux

You will find that you will simply not be able to port every user over to Linux at all times. Rather than concede defeat, you have two alternatives:

- **Use an emulator.** Install software that allows your Linux system to imitate, as it were, a Windows system. Once you get the emulator properly installed, all you have to do is install a Windows application on the Linux system. The emulator will then allow the Windows application to run directly from inside Linux.
- **Use remote desktop administration software.** Simply install a server that allows you to directly access the desktop through a Web browser or specialized application.

Following is a discussion of each option.

Compatibility Layer Software

In many ways, the software discussed here is not emulation software. In the strictest sense, emulator software recreates the software application programming interfaces (APIs), and the actual functions of the CPU (for example, a Pentium chip). Wine, CrossOver Office, and Win 4 Lin Workstation do not recreate the architecture of the CPU. Therefore, they are technically not emulators.

Nevertheless, it is still common practice to lump this software into the emulator category, because using applications such as Wine, you can make your Linux system behave as if it was a Windows system. In fact, if you properly configure these applications, certain native Windows applications will run, thinking that they are in a Windows environment. These applications use sets of APIs to help

convince native Windows applications that they are, in fact, running on Windows.

So, to avoid controversy, we will not call these applications “emulators,” even though that’s basically what they are. Taking the lead of the developers of Wine, we are calling these applications “compatibility layer software,” because they all create a layer between the Linux operating system and the Windows application.

The benefit of this type of emulator, well, software, is that you can use native Windows applications directly from your Linux desktop. You do not have to rely on a network connection to another system. However, emulators can be somewhat tricky to configure, and the slightest change in the application’s configuration can “break” your configuration and force a time-consuming and possibly costly service call.

As you prepare to use an emulator, ask the following questions:

- What version of the Windows operating system does the application require?
- Do you require access to raw data from inside Linux?
- How many people need to access these applications, and the resulting data from them, at one time? In short, what is the expected load on this system?

These questions will help you determine the correct hardware size, and the appropriate software. Now, let’s look at some of the common emulators available.

Wine

Wine is an acronym for “Wine is not an emulator.” Wine is meant to provide a replacement for Windows; it does not require Windows to run. Therefore, you do not need a Windows license to run a Windows application. You will, however, need a license to run the application. Suppose, for example, that you managed to run Microsoft Word on Wine. You would not need a license for the Microsoft Windows operating system. However, you would need to license Microsoft Word.

It is important to understand that Wine has enjoyed a “work in progress” standing for many years. Many Windows applications do run in Wine. A list of Windows applications verified to run in Wine is available at www.winehq.org/site/supported_applications.

A Web site called “Frank’s Corner” (<http://frankscorner.org>) provides tips to help get various applications going. Applications that Frank has worked with include:

- Microsoft Office 2000
- Macromedia Flash MX
- PhotoShop 7.0

People have had significant success with Wine. However, Wine is not yet a “production quality” tool; it is more of an extended “hack in motion.” The fact that your needed application runs today on the latest and greatest version of Wine is no guarantee that it will run properly when you upgrade to the next version. However, there is a much more reliable application: Code Weavers’ CrossOver Office.

Code Weavers’ CrossOver Office

CrossOver Office is essentially a perfected, commercial version of Wine. CrossOver Office allows any Windows application to run smoothly (or, as smoothly as any application can run using compatibility software). As with Wine, if you use CrossOver Office you do not need to purchase a Windows license. You will find that with CrossOver Office, upgrades will not cause existing configurations to fail. In addition, CrossOver Office makes it possible to run all of the Visual Basic macros on which many Microsoft Office users rely.

CrossOver Office makes it relatively easy to install and run Windows applications in Linux. Still, there are drawbacks to this solution. First, CrossOver Office requires significant amounts of memory. In addition, not all of the features of your Windows applications will be available. Therefore, although you may be able to run a copy of Macromedia Flash MX, you may still find some features missing.

In spite of these drawbacks, you will likely find that between the alternative programs discussed previously and applications such as CrossOver Office, you will be able to migrate any user to Linux. To learn more about CrossOver Office, go to www.codeweavers.com/site/products.

Summary

Choosing the appropriate desktop environment requires several skills. First, you need to know about the options. Second, you need to identify what the customer wants and needs. You then need to know how to match current technologies to those customer needs. In this chapter, you learned about available technologies, and how to weigh them against customer needs.

From common desktops such as Gnome and KDE to e-mail and Web applications, you learned how to recommend solutions that can save end users time and money. You also learned how to migrate settings, and how to install native applications on Linux that cannot, for some reason, be replaced by their Linux counterparts.

This chapter helped you identify problems, possibilities, and solutions. Now that you are more familiar with Linux desktop solutions, continue your learning process by installing some of the software profiled in this chapter. The only way you can take the next step in your knowledge and ability to solve problems is to go through the process of installing the software.

Solutions Fast Track

Common Desktop Environments

- ☑ The two most popular desktop environments are Gnome and KDE.
- ☑ Gnome and KDE have significant similarities.
- ☑ It is your responsibility to weigh the relative merits and benefits.
- ☑ Let your clients view the desktop environments to make an informed choice.

The X Window System and Window Managers

- ☑ An X Window server is responsible for providing resources to the window manager, which acts as a client.
- ☑ You can choose between several window managers.
- ☑ A window manager is not the same thing as a desktop environment; a desktop environment contains a window manager.

- ☑ Window managers help determine the look and feel of the elements provided by a desktop environment, including how elements are positioned, the look and feel of toolbars, and fonts.

E-Mail and Personal Information Management Clients

- ☑ Many e-mail clients exist. Some are “straight” e-mail clients, whereas others have access to groupware features, including calendaring and scheduling.
- ☑ Migrating e-mail settings might require you to adopt an intermediate strategy, where you import settings into one browser, and then into the next. This strategy helps you get information into a format that the destination e-mail browser can understand.
- ☑ If all else fails, you can migrate messages by simply forwarding them from one machine to the next.

Web Browsers

- ☑ It is likely that no one Web browser will meet everyone’s needs.
- ☑ Browsers are now used to access information of all types.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- ☑ The latest versions of Mozilla have the capability to conduct Microsoft CHAP-based sessions.

Office Application Suites

- ☑ To remain productive, end users will have to be provided software that helps them fulfill their duties.
- ☑ It is your responsibility to be as straightforward as possible in regard to the capabilities of office suites and applications available in Linux. Do not exaggerate. Give end users a “trial run” so they know exactly what they are getting into.
- ☑ OpenOffice.org software may appear to be the best fit, because it is feature-rich and free. However, in some cases, OpenOffice.org and even StarOffice will not suffice. In these cases, you may want to purchase a tool such as Hancom Office. You may even need to research how to run native Windows applications in Linux.

Running Windows Applications on Linux

- ☑ Software such as Wine (www.winehq.org) is not strictly emulation software. True emulation software emulates the necessary software libraries to run, and the actual CPU (for example, a Pentium processor). Wine simply contains the application programmer interface (API) software necessary to “trick” the native Windows applications into thinking that they are running on a Windows system.
- ☑ When working with emulators, remember to ask at least three questions: First, “What version of the Windows operating system does the application require?” Second, “Do you require access to raw data

from inside Linux itself?” Third, determine what is expected of the system by asking “How many people need to access these applications, and the resulting data from them, at one time? In short, what is the expected load on this system?”

- ☑ Code Weavers’ CrossOver Office is a commercial version of Wine. Therefore, it is much more reliable. Understand, though, that even CrossOver Office has limitations. For example, although a Windows application may run, it is possible that not all features will be activated.

Q: Which is better, KDE or Gnome?

A: Study both interfaces, and the demands of your customers, to determine the answer. Consider issues such as performance, flexibility, and application availability. It is possible to make the interfaces resemble each other. It is also possible to customize these environments so that they look radically different from each other.

Q: Can Gnome applications run on KDE and Blackbox environments?

A: Yes, for the most part. There may be exceptions, but Evolution, for example, will work equally well in KDE and Gnome. KMail works equally well in Gnome and Blackbox.

Q: I have decided to have my employees use three different browsers. Do I have to install RealPlayer three different times?

A: No. Simply access the necessary menus to direct the browser to find the RealPlayer binary (realplay). You may also have to define a MIME type.

Q: I have had requests from department managers who want their employees to have the necessary software to create PDFs. They do not want to purchase Adobe Acrobat. Which application can I use to create PDF files?

A: There are several, including OpenOffice.org, StarOffice, Hancom Office, and CrossOver Office.

Q: Why is Code Weavers’ CrossOver Office more stable than Wine? I thought they were similar.

A: Code Weavers’ CrossOver Office is a commercial product with a dedicated team that works daily on software improvements. The Wine team does not have the same amount of time available.

Security Appendixes

The first eleven chapters of this book have provided you with the information and tools to migrate your network from a Windows to Linux environment. These Security Appendixes will now provide you with an introduction to some of the most important information and open source tools you can now use to monitor and secure your network. For more in depth information on the tools covered in these appendixes, please refer to:

Ethereal Packet Sniffing by Angela Orebaugh and Gilbert Ramirez (Syngress ISBN: 1-932266-82-8).

Snort 2.1 Intrusion Detection, Second Edition by Brian Caswell and Jay Beale (Syngress ISBN 1-931836-04-3).

Nessus Network Auditing by Renaud Deraison, HD Moore, and Jay Beale (Syngress ISBN 1-931836-08-6).

Introducing Network Analysis and Ethereal

Solutions in this Appendix:

- What is Network Analysis and Sniffing?
- Who Uses Network Analysis?
- How Does it Work?
- Detecting Sniffers
- Protecting Against Sniffers
- Network Analysis and Policy

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

“Why is the network slow?” “Why can’t I access my e-mail?” “Why can’t I get to the shared drive?” “Why is my computer acting strange?” If you are a systems administrator, network engineer, or security engineer you have probably heard these questions countless times. Thus begins the tedious and sometimes painful journey of troubleshooting. You start by trying to replicate the problem from your computer. Sure enough, you can’t get to anything on the local network or the Internet either. Now what? Go to each of the servers and make sure they are up and functioning? Check that your router is functioning? Check each computer for a malfunctioning network card?

What about this scenario: you go to your main access switch, or border router, and configure one of the unused ports for port mirroring. You plug in your laptop, fire up your network analyzer, and see thousands of User Datagram Protocol (UDP) packets destined for port 1434 with various, apparently random, Internet Protocol (IP) addresses. You immediately apply access filters to block these packets from entering or exiting your network until you do more investigating. A quick search on the Internet holds the answer. The date is January 25, 2003, and you have just been hit with the SQL Slammer worm. You were able to contain the problem relatively quickly thanks to your knowledge and use of your network analyzer.

What is Network Analysis and Sniffing?

Network analysis is the process of capturing network traffic and inspecting it closely to determine what is happening on the network. A network analyzer decodes, or dissects, the data packets of common protocols and displays the network traffic in human-readable format. Network analysis is also known by several other names: traffic analysis, protocol analysis, sniffing, packet analysis, and eavesdropping to name a few. Sniffing tends to be one of the most popular terms in use today. However, as you will see later in this appendix, due to malicious users it has had a negative connotation in the past.

A network analyzer can be a standalone hardware device with specialized software, or it can simply be software that you install on your desktop or laptop computer. Network analyzers are available both free and commercially. Differences between network analyzers tend to depend on features such as the number of supported protocol decodes, the user interface, and graphing and statistical capabilities. Other differences include inference capabilities, such as expert

analysis features, and the quality of packet decodes. Although several network analyzers all decode the same protocols, some may decode better than others.

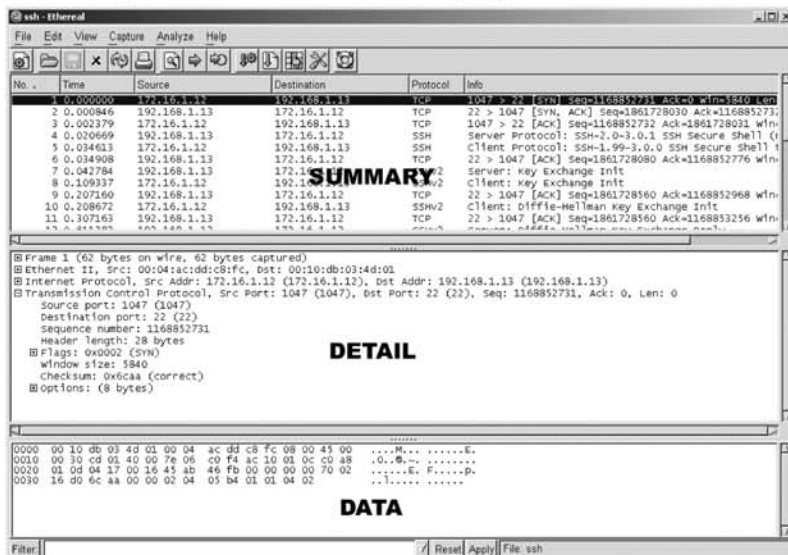
NOTE

Sniffer (with a capital “S”) is a trademark owned by Network Associates referring to its Sniffer product line. However, it has become common industry usage that a “sniffer” (with a lower case “s”) is a program that captures and analyzes network traffic.

Figure A.1 shows the Ethereal Network Analyzer display windows. A typical network analyzer displays the captured traffic in three panes:

- **Summary** This pane displays a one line summary of the capture. Fields usually include date, time, source address, destination address, and the name and information about the highest-layer protocol.
- **Detail** This pane provides all of the details for each of the layers contained inside the captured packet in a tree-like structure.
- **Data** This pane displays the raw captured data both in hexadecimal and ASCII format.

Figure A.1 Example Network Analyzer Display



A network analyzer is a combination of hardware and software. Although there are differences in each product, a network analyzer is composed of five basic parts:

- **Hardware** Most network analyzers are software-based and work with standard operating systems (OSs) and network interface cards (NICs). However, there are some special hardware network analyzers that offer additional benefits such as analyzing hardware faults including: Cyclic Redundancy Check (CRC) errors, voltage problems, cable problems, jitter, jabber, negotiation errors, etc. Some network analyzers only support Ethernet or wireless adapters, while others support multiple adapters and allow users to customize their configuration. Sometimes you will also need a hub or a cable tap to connect to the existing cable.
- **Capture driver** This is the part of a network analyzer that is responsible for actually capturing the raw network traffic from the cable. It will also filter out the traffic that you want and store the data in a buffer. This is the core of a network analyzer and you cannot capture data without it.
- **Buffer** This component stores the captured data. Data can be stored in a buffer until it is full, or in a rotation method such as “round robin” where the newest data replaces the oldest data. Buffers can be disk-based or memory-based.
- **Real-time analysis** This feature analyzes the data as it comes off the cable. Some network analyzers use this to find network performance issues, and network intrusion detection systems do this to look for signs of intruder activity.
- **Decode** This component displays the contents of the network traffic with descriptions so that it is human-readable. Decodes are specific to each protocol, so network analyzers tend to vary in the number of decodes they currently support. However, new decodes are constantly being added to network analyzers.

NOTE

Jitter is a term used to describe the random variation in the timing of a signal. Electromagnetic interference and crosstalk with other signals can cause jitter. *Jabber* is when a device is improperly handling electrical signals, thus affecting the rest of the network. Faulty network interface cards can cause jabber.

Who Uses Network Analysis?

System administrators, network engineers, security engineers, system operators, even programmers, all use network analyzers. Network analyzers are invaluable tools for diagnosing and troubleshooting network problems. Network analyzers used to be dedicated hardware devices that were very expensive. New advances in technology have allowed for the development of software network analyzers. This makes it more convenient and affordable for administrators to effectively troubleshoot a network. It also brings the capability of network analysis to anyone who wishes to perform it.

The art of network analysis is a double-edged sword. While network, system, and security professionals use it for troubleshooting and monitoring of the network, intruders can also use network analysis for harmful purposes. A network analyzer is a tool, and like all tools they can be used for both good and bad intentions.

The following list describes a few reasons why administrators use network analyzers:

- Converting the binary data in packets to human-readable format
- Troubleshooting problems on the network
- Analyzing the performance of a network to discover bottlenecks
- Network intrusion detection
- Logging network traffic for forensics and evidence
- Analyzing the operations of applications
- Discovering a faulty network card
- Discovering the origin of a Denial of Service (DoS) attack

- Detecting spyware
- Network programming to debug in the development stage
- Detecting a compromised computer
- Validating compliance with company policy
- As an educational resource when learning about protocols
- For reverse-engineering protocols in order to write clients and supporting programs

How are Intruders Using Sniffers?

When used by malicious individuals, sniffers can represent a significant threat to the security of your network. Network intruders often use network sniffing to capture valuable, confidential information. The terms sniffing and eavesdropping have often been associated with this practice. However, sniffing is now becoming a non-negative term and most people use the terms sniffing and network analysis interchangeably.

Using a sniffer in an illegitimate way is considered a passive attack. It does not directly interface or connect to any other systems on the network. However, the computer that the sniffer is installed on could have been compromised using an active attack. The passive nature of sniffers is what makes detecting them so difficult. We will discuss the methods used to detect sniffers later in this appendix.

The following list describes a few reasons why intruders are using sniffers on the network:

- Capturing clear-text usernames and passwords
- Compromising proprietary information
- Capturing and replaying Voice over IP telephone conversations
- Mapping a network
- Passive OS fingerprinting

Obviously, these are illegal uses of a sniffer, unless you are a penetration tester whose job it is to find these types of weaknesses and report them to an organization.

For sniffing to occur, an intruder must first gain access to the communication cable of the systems that are of interest. This means being on the same shared net-

work segment, or tapping into the cable somewhere between the path of communications. If the intruder is not physically present at the target system or communications access point, there are still ways to sniff network traffic. These include:

- Breaking into a target computer and installing remotely controlled sniffing software.
- Breaking into a communications access point, such as an Internet Service Provider (ISP) and installing sniffing software.
- Locating/finding a system at the ISP that already has sniffing software installed.
- Using social engineering to gain physical access at an ISP to install a packet sniffer.
- Having an insider accomplice at the target computer organization or the ISP install the sniffer.
- Redirecting communications to take a path that includes the intruder's computer.

Sniffing programs are included with most *rootkits* that are typically installed on compromised systems. Rootkits are used to cover the tracks of the intruder by replacing commands and utilities and clearing log entries. They also install other programs such as sniffers, key loggers, and backdoor access software. Windows sniffing can be accomplished as part of some RAT (Remote Admin Trojan) such as SubSeven or Back Orifice. Often intruders will use sniffing programs that are configured to detect specific things, such as passwords, and then electronically send them to the intruder (or store them for later retrieval by the intruder). Vulnerable protocols for this type of activity include telnet, FTP, POP3, IMAP, SMTP, HTTP, rlogin, and SNMP.

One example of a rootkit is T0rnKit, which works on Solaris and Linux. The sniffer that is included with this rootkit is called t0rns and is installed in the hidden directory `/usr/srec/.puta`. Another example of a rootkit is Lrk5 (Linux Rootkit 5), which installs with the linsniff sniffer.

Intruders commonly use sniffer programs to control back doors. One method is to install a sniffer on a target system that listens for specific information. Then, backdoor control information can be sent to a neighboring system. The sniffer picks this up, and acts appropriately on the target computer. This type of backdoor control is often hard for investigators to detect, since it looks like the innocent neighbor system is the compromised target.

cd00r is an example of a backdoor sniffer that operates in non-promiscuous mode, making it even harder to detect. Using a product like Nmap to send a series of Transmission Control Protocol (TCP) SYN packets to several predefined ports will trigger the backdoor to open up on a pre-configured port. More information about Cdoor can be found at www.phenoelit.de/stuff/cd00r.c.

NOTE

A *rootkit* is a collection of trojan programs that are used to replace the real programs on a compromised system in order to avoid detection. Some common commands that get replaced are *ps*, *ifconfig*, and *ls*. Rootkits also install additional software such as sniffers.

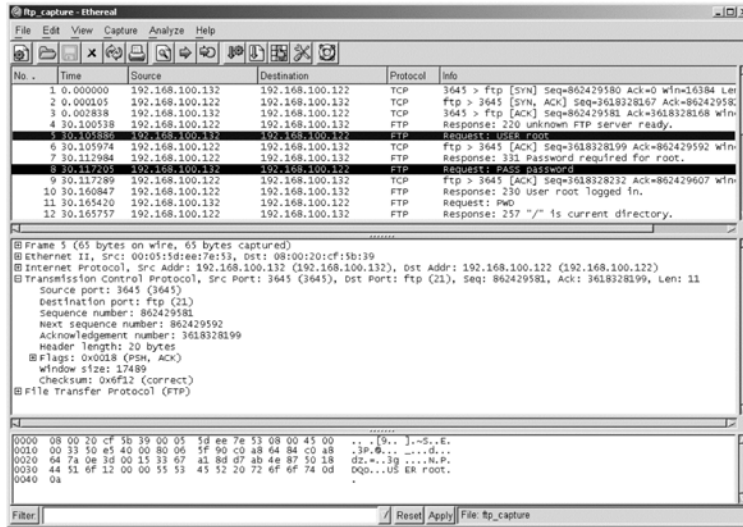
NOTE

Nmap is a network scanning tool used for network discovery and auditing. It can send raw IP packets to destination ports on target systems.

What does Sniffed Data Look Like?

We have done a lot of talking about sniffers and what they are used for, but the easiest way to grasp the concepts previously discussed is watching a sniffer in action. Figure A.2 shows a capture of a simple FTP session from a laptop to a Sun Solaris system. The two highlighted packets show you just how easy it is to sniff the username and password. In this case, the username is “root” and the password is “password”. Of course, allowing root to FTP into a system is a very poor security practice; this is just for illustration purposes!

Figure A.2 Example of Sniffing a Connection



Common Network Analyzers

A simple search on SecurityFocus (www.securityfocus.org/tools/category/4) shows the diversity and number of sniffers available. Some of the most prominent ones are:

- **Ethereal** Of course, this one is the topic of this book! Ethereal is obviously one of the best sniffers available. It is being developed as a free commercial quality sniffer. It has numerous features, a nice graphical user interface (GUI), decodes for over 400 protocols, and it is actively being developed and maintained. It runs on both UNIX-based systems and Windows. This is a great sniffer to use, even in a production environment. It is available at www.ethereal.com.
- **WinDump** This is the Windows version of tcpdump available at <http://windump.polito.it>. It uses the WinPcap library and runs on Windows 95/98/ME/NT/2000/XP.
- **Network Associates Sniffer** This is one of the most popular commercial products available. Now marketed under McAfee Network Protection Solutions, Network Associates has an entire Sniffer product line for you to peruse at www.nai.com.

- **Windows 2000/NT Server Network Monitor** Both Windows 2000 Server and NT Server have a built-in program to perform network analysis. It is located in the Administrative tools folder, but is not installed by default, so you may have to add it from the installation CD.
- **EtherPeek** This is a commercial network analyzer by WildPackets. There are versions for both Windows and Mac, as well as other network analysis products that can be found at www.wildpackets.com.
- **Tcpdump** This is the oldest and most common network sniffer. The Network Research Group (NRG) of the Information and Computing Sciences Division (ICSD) at Lawrence Berkeley National Laboratory (LBNL) developed tcpdump. It is command line-based and runs on UNIX-based systems. It is being actively developed and maintained at www.tcpdump.org.
- **Snoop** This command line network sniffer is included with the Sun Solaris operating system. It is especially competent at decoding Sun-specific protocols.
- **Sniffit** This network sniffer runs on Linux, SunOS, Solaris, FreeBSD and IRIX. It is available at <http://reptile.rug.ac.be/~coder/sniffit/sniffit.html>.
- **Snort** This is a network intrusion detection system that uses network sniffing. It is actively developed and maintained at www.snort.org. For more information, refer to *Snort 2.0: Intrusion Detection* (Syngress Publishing, ISBN: 1-931836-74-4)
- **Dsniff** This is very popular network sniffing package. It is a collection of programs to sniff specifically for interesting data such as passwords, and to facilitate the sniffing process such as evading switches. It is actively maintained at www.monkey.org/~dugsong/dsniff.
- **Ettercap** This sniffer is designed specifically to sniff in a switched network. It has built-in features such as password collecting, OS fingerprinting, and character injection. It runs on several platforms including Linux, Windows, and Solaris. It is actively maintained at <http://ettercap.sourceforge.net>.
- **Analyzer** This is a free sniffer for the Windows OS that is being actively developed by the makers of WinPcap and WinDump at

Politecnico di Torino. It can be downloaded from <http://analyzer.polito.it>.

- **Packetizer** This is a free sniffer for the Windows OS that uses Ethereal's core logic. It tends to run a version or two behind the current release of Ethereal. It is actively maintained by Network Chemistry at www.networkchemistry.com/products/packetizer/index.html.

Notes from the Underground...

Carnivore or Vegetarian?

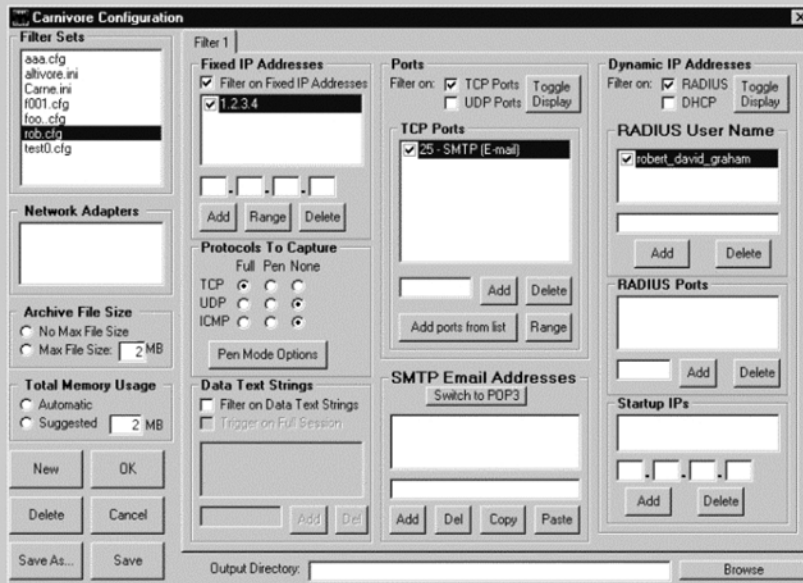
No talk about network analyzers would be complete without the mention of Carnivore. While certainly not a commonly used network analyzer, it has created a lot of talk in the security world as well as the media. Carnivore is the code name for the FBI's network analyzer. It is used to monitor relevant communications among selected individuals as part of a criminal investigation. Its name has been changed to DCS100 in an attempt to obscure its image and to calm the public's fear of its misuse. When necessary, federal agents will arrive at a suspect's ISP with a "black box", which is merely a dedicated server running Windows 2000 or NT and the FBI's Carnivore software preloaded. The server is placed on the ISP's trunk to read header information for any traffic going to or coming from the suspect. This was rather common at numerous ISPs after September 11, 2001.

Many people have been concerned about the use of Carnivore and its ability to intercept all traffic, mostly due to privacy issues. They are concerned about how Carnivore works, how it could be misused by law enforcement, and the privacy debate over cable taps in general.

Carnivore is an Internet wiretap designed by the U.S. Federal Bureau of Investigation (FBI). It is designed with the special needs of law enforcement in mind. For example, some court orders might allow a pen-register monitoring of just the From/To e-mail addresses, whereas other court orders might allow a full capture of the e-mail. A summary of Carnivore's features can be seen within the configuration program shown in Figure A.3.

Continued

Figure A.3 Carnivore Configuration Program



The features are:

- **Filter sets** The settings are saved in configuration files; the user can quickly change the monitoring by selecting a different filter set.
- **Network adapters** A system may have multiple network adapters; only one can be selected for sniffing at a time.
- **Archive file size** A limit can be set on how much data is captured; by default, it fills up the disk.
- **Total memory usage** Network traffic may come in bursts faster than it can be written to disk; memory is set aside to buffer the incoming data.
- **Fixed IP address** All traffic to/from a range of IP addresses can be filtered. For example, the suspect may have a fixed IP address of 1.2.3.4 assigned to their cable modem. The FBI might get a court order allowing them to sniff all of the suspect's traffic.

Continued

- **Protocols to capture** Typically, a court order will allow only specific traffic to be monitored, such as SMTP over TCP. In Pen mode, only the headers are captured.
- **Data text strings** This is the Echelon feature that looks for keywords in traffic. A court order must specify exactly what is to be monitored, such as an IP address or e-mail account. Such wide-open keyword searches are illegal in the United States. The FBI initially denied that Carnivore had this feature.
- **Ports** A list of TCP and UDP ports can be specified. For example, if the FBI has a court order allowing e-mail capture, they might specify the e-mail ports of 25 (SMTP), 110 (POP3), and 143 (IMAP).
- **SMTP e-mail addresses** A typical scenario is where Carnivore monitors an ISP's e-mail server, discarding all e-mails except those of the suspects. An e-mail session is tracked until the suspect's e-mail address is seen, then all the packets that make up the e-mail are captured.
- **Dynamic IP addresses** When users dial up the Internet, they are logged in via the RADIUS protocol, which then assigns them an IP address. Normally, the FBI will ask the ISP to reconfigure their RADIUS servers to always assign the same IP address to the suspect, and will then monitor all traffic to/from that IP address. Note: if you are a dial-up user and suspect the FBI is after you, check to see if your IP address is the same every time you dial up. Sometimes this isn't possible. Carnivore can be configured to monitor the RADIUS protocol and dynamically discover the new IP address assigned to the suspect. Monitoring begins when the IP address is assigned, and stops when it is unassigned.

The FBI developed Carnivore because other existing utilities do not meet the needs of law enforcement. When an e-mail is sent across the wire, it is broken down into multiple packets. A utility like mailsnarf will reassemble the e-mail back into its original form. This is bad because the suspect's defense attorneys will challenge its accuracy: Did a packet get dropped somewhere in the middle that changes the meaning of the e-mail? Did a packet from a different e-mail somehow get inserted into the message? By capturing the raw packets rather than reassembling

Continued

them, Carnivore maintains the original sequence numbers, ports, and timestamps. Any missing or extra packets are clearly visible, allowing the FBI to defend the accuracy of the system.

Another problem that the FBI faces is minimization of the sniffed data. When the FBI wiretaps your line, they must assign an agent to listen in. If somebody else uses your phone (like your spouse or kids), they are required to turn off the tape recorders. In much the same way, Carnivore is designed to avoid capturing anything that does not belong to the suspect. A typical example would be using Carnivore to monitor the activities of a dial-up user. Carnivore contains a module to monitor the RADIUS traffic that is used by most ISPs to authenticate the user and assign a dynamic IP address. This allows Carnivore to monitor only that user without intercepting any other traffic.*

The following websites have more information on Carnivore:

- www.fbi.gov
- www.robertgraham.com/pubs/carnivore-faq.html
- www.stopcarnivore.org

*Excerpt from Robert Graham's chapter in *Hack Proofing Your Network, Second Edition*. Syngress Publishing 1-928994-70-9.

How Does It Work?

This section provides an overview of how all of this sniffing takes place. It gives you a little background on how networks and protocols work; however, there are many excellent resources out there that fill entire books themselves! The most popular and undoubtedly one of the best resources is Richard Stevens' "TCP/IP Illustrated, Vol. 1 – 3".

Explaining Ethernet

Ethernet is the most popular protocol standard used to enable computers to communicate. A protocol is like speaking a particular language. Ethernet was built around a principle of a shared medium where all computers on the local network segment share the same cable. It is known as a *broadcast* protocol because when a computer has information to send, it sends that data out to all other computers on the same network segment. This information is divided up into

manageable chunks called packets. Each packet has a header, which is like an envelope containing the addresses of both the destination and source computers. Even though this information is sent out to all computers on a segment, only the computer with the matching destination address will respond. All of the other computers on the network still see the packet, but if they are not the intended receiver they will disregard and discard it, unless a computer is running a sniffer. When you are running a sniffer, the packet capture driver that we mentioned earlier will put the computer's NIC into what is known as promiscuous mode. This means that the sniffing computer will be able to see all of the traffic on the segment regardless of who it is being sent to. Normally computers run in non-promiscuous mode, listening for information only designated for themselves. However, when a NIC is in promiscuous mode it can see conversations to and from all of its neighbors.

Ethernet addresses are known as Media Access Control (MAC) addresses, hardware addresses, or sometimes just Ethernet addresses. Since many computers may share a single Ethernet segment, each must have an individual identifier. These identifiers are hard-coded on to the NIC. A MAC address is a 48-bit number, also stated as a 12-digit hexadecimal number. This number is broken down into two halves, the first 24-bits identify the vendor of the Ethernet card, and the second 24-bits is a serial number assigned by the vendor.

The following steps will allow you to view your NIC's MAC address:

- **Windows 9x** Access **Start | Run**, and type **winipcfg.exe**. The MAC address will be listed as "Adapter Address".
- **Windows NT/2000/XP** Access the command line and type **ipconfig /all**. The MAC address will be listed as "Physical Address".
- **Linux and Solaris** Type **ifconfig -a** at the command line. The MAC address will be listed as "HWaddr" on Linux and "ether" on Solaris.

You can also view the MAC addresses of other computers that you have communicated with recently, by using the command **arp -a**. More will be discussed about this in the "Defeating Switches" section.

MAC addresses are unique, and no two computers should have the same one. However, this is not always the case. Occasionally there could be a manufacturing error that would cause more than one network interface card to have the same MAC address, but mostly, people will change their MAC addresses on purpose. This can be done with a program, such as **ifconfig**, that will allow you to fake your MAC address. Faking your MAC address is also called *spoofing*. Also, some

adapters allow you to use a program to reconfigure the runtime MAC address. And lastly with the right tools and skill you can physically re-burn the address into the network interface card.

NOTE

Spoofing is the altering of network packet information such as the IP source address, MAC address, or even an e-mail address. This is often done to masquerade as another device in order to exploit a trust relationship, or to make tracing the source of attacks difficult. Address spoofing is also used in denial of service (DoS) attacks, such as Smurf, where the return address of network requests are spoofed to be the IP address of the victim.

Understanding the OSI model

The International Standards Organization (ISO) developed the Open Systems Interconnection (OSI) model in the early 1980's to describe how network protocols and components work together. It divides network functions into seven layers, and each layer represents a group of related specifications, functions, and activities.

The layers of the OSI model are:

- **Application layer** This topmost layer of the OSI model is responsible for managing communications between network applications. This layer is not the application program itself, although some applications may have the ability and the underlying protocols to perform application layer functions. For example, a Web browser is an application, but it is the underlying Hypertext Transfer Protocol (HTTP) protocol that provides the application layer functionality. Examples of application layer protocols include File Transfer Protocol (FTP), Simple Network Management Protocol (SNMP), Simple Mail Transfer Protocol (SMTP), and Telnet.
- **Presentation layer** This layer is responsible for data presentation, encryption, and compression.

- **Session layer** The session layer is responsible for creating and managing sessions between end systems. The session layer protocol is often unused in many protocols. Examples of protocols at the session layer include NetBIOS and Remote Procedure Call (RPC).
- **Transport layer** This layer is responsible for communication between programs or processes. Port or socket numbers are used to identify these unique processes. Examples of transport layer protocols include: TCP, UDP, and Sequenced Packet Exchange (SPX).
- **Network layer** This layer is responsible for addressing and delivering packets from the source computer to the destination computer. The network layer takes data from the transport layer and wraps it inside a packet or datagram. Logical network addresses are generally assigned to computers at this layer. Examples of network layer protocols include IP and Internetwork Packet Exchange (IPX). Devices that work at this layer are routers and Layer 3 switches.
- **Data link layer** This layer is responsible for delivering frames between NICs on the same physical segment. Communication at the data link layer is generally based on MAC addresses. The data link layer wraps data from the network layer inside a frame. Examples of data link layer protocols include Ethernet, Token Ring, and Point-to-Point Protocol (PPP). Devices that operate at this layer include bridges and switches.
- **Physical layer** This layer defines connectors, wiring, and the specifications on how voltage and bits pass over the cabled or wireless media. Devices at this layer include repeaters, concentrators, hubs, and cable taps. Devices that operate at the physical layer do not have an understanding of network paths.

NOTE

The terms *frame* and *packet* tend to be used interchangeably when talking about network traffic. However, the difference lies in the various layers of the OSI model. A frame is a unit of transmission at the data link layer. A packet is a unit of transmission at the network layer, however many people use the term packet to refer to data at any layer.

The OSI model is very generic and can be used to explain virtually any network protocol. Various protocol suites are often mapped against the OSI model for this purpose. A solid understanding of the OSI model aids tremendously in network analysis, comparison, and troubleshooting. However, it is also important to remember that not all protocols map nicely to the OSI model. For example, TCP/IP was designed to map to the U.S. Department of Defense (DoD) model. In the 1970s, the DoD developed its four-layer model. The core Internet protocols adhere to this model.

The DoD model is merely a condensed version of the OSI model. Its four layers are:

- **Process layer** This layer defines protocols that implement user-level applications such as mail delivery, remote login, and file transfer.
- **Host-to-host layer** This layer handles the connection, data flow management, and retransmission of lost data.
- **Internet layer** This layer is responsible for delivering data from source host to destination host across a set of different physical networks that connect the two machines.
- **Network access layer** This layer handles the delivery of data over a particular hardware media.

Notes from the Underground...

The TCP/IP Protocols

You will be seeing a lot of references in this book to TCP/IP and its associated protocols, specifically IP, TCP, and UDP. TCP/IP, developed by the Defense Advanced Research Projects Agency (DARPA), is the most widely used routed protocol today. IP is a Layer 3 protocol that contains addressing and control information that allows packets to be routed. IP is a connectionless protocol; therefore, it provides unreliable best-effort packet delivery service. Since IP only provides best-effort delivery, a packet may be discarded during transmission. All IP packets consist of a header and a payload (data from upper layers).

At the transport layer of the TCP/IP stack, the two commonly used protocols are TCP and UDP. The headers for both of these protocols

Continued

include a source and destination port number, which are used to determine the application or process that the TCP segment or UDP datagram originate from and destined to. TCP is a connection-oriented protocol, and UDP is a connectionless protocol. The TCP header includes sequence and acknowledgment numbers for reliable delivery. When IP needs reliable, guaranteed transfers it depends on TCP to provide this functionality.

Since TCP is a connection-oriented protocol it creates a dialog between the two communicating hosts to establish a connection. This is known as the three-way handshake. It starts by Host A sending a *SYN* packet to Host B letting it know that it wants to talk. Host B then responds with a *SYN/ACK*, saying that it is available to talk. Host A then finalizes the connection with an *ACK*.

TCP can also use the sliding window principle. The sliding window algorithm allows a buffer to be placed between the application program and the network data flow. Data received from the network is placed into this buffer until the application is ready to read it. The window is the amount of data that can be fetched into the buffer before an acknowledgment must be sent. Examples of applications that use TCP include FTP, Telnet, Network File System (NFS), SMTP, HTTP, Domain Name System (DNS), and Network News Transfer Protocol (NNTP). Examples of applications that use UDP include DNS, Routing Information Protocol (RIP), NFS, SNMP, and Dynamic Host Configuration Protocol/Boot Protocol (DHCP/BOOTP). As you can see, some applications (such as DNS and NFS) can use both protocols.

Notes from the Underground...

Writing Your Own Sniffer

There is an excellent paper titled "Basic Packet-Sniffer Construction from the Ground Up" by Chad Renfro located at www.unixgeeks.org/security/newbie/security/sniffer/sniffer_construction.txt. In this paper he presented a very basic 28-line packet sniffer written in C, called *sniff.c*. Even if you aren't a programmer, Chad explains the program line by line in an

Continued

easy to understand manner. The program demonstrates the use of the `RAW_SOCKET` device to read TCP packets from the network and print basic header information to `std_out`. For simplicity, the program operates in non-promiscuous mode, so you would first need to put your interface in promiscuous mode by using the **`ifconfig eth0 promisc`** command.

There is also a header file that has to be copied into the same directory as `sniff.c`. It provides standard structures to access the IP and TCP fields. The structures identify each field in the IP and TCP header. It contains more information than what the `sniff.c` actually uses, but it least it is there to build upon.

To run the program, copy the `sniff.c` and `headers.h` into the same directory, and enter the command **`gcc -o sniff sniff.c`**. This will compile the program and create an executable file called `sniff`, which can be run by typing `./sniff`. The following text shows the output of the sniff program when I attempted a TELNET and FTP connection:

```
Bytes received :::      48
Source address ::: 192.168.1.1
IP header length ::: 5
Protocol ::: 6
Source port ::: 1372
Dest port  ::: 23
Bytes received :::      48
Source address ::: 192.168.1.1
IP header length ::: 5
Protocol ::: 6
Source port ::: 1374
Dest port  ::: 21
```

Once you are done capturing data, you can end the program by typing **CTRL-C**. You may also want to remove your interface from promiscuous mode by typing the command **`ifconfig eth0 -promisc`**.

CSMA/CD

Ethernet uses the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol for devices on the network to exchange data. The term

multiple access refers to the fact that many network devices attached to the same segment have the opportunity to transmit. Each device is given an equal opportunity; no device has priority over any other. *Carrier sense* describes how an Ethernet interface on a network device listens to the cable before transmitting. The network interfacer ensures that there are no other signals on the cable before it transmits. An Ethernet interface also listens while transmitting to ensure that no other network device transmits data at the same time. When two network devices transmit at the same time, a *collision* occurs. Since Ethernet interfaces listen to the media while they are transmitting, they are able to identify the presence of others through their *collision detection* method. If a collision occurs, the transmitting device will wait a random amount of time before retransmitting. This function is known as *random backoff*.

Traditionally, Ethernet operation has been *half duplex*. This means that an interface may either transmit or receive data, but it cannot do both at the same time. If more than one network interface on a segment tries to transmit at the same time, a collision occurs, as per CSMA/CD. When a crossover cable is used to connect two devices or a single device is attached to a switch port, only two interfaces on the segment need to transmit or receive and no collisions occur. This is because the transmit (TX) of device A is connected to the receive (RX) of device B, and the TX of B is connected to the RX of A. The collision detection method is therefore no longer necessary, so interfaces can be placed in *full-duplex* mode of operation. This mode allows network devices to transmit and receive at the same time, thereby increasing performance.

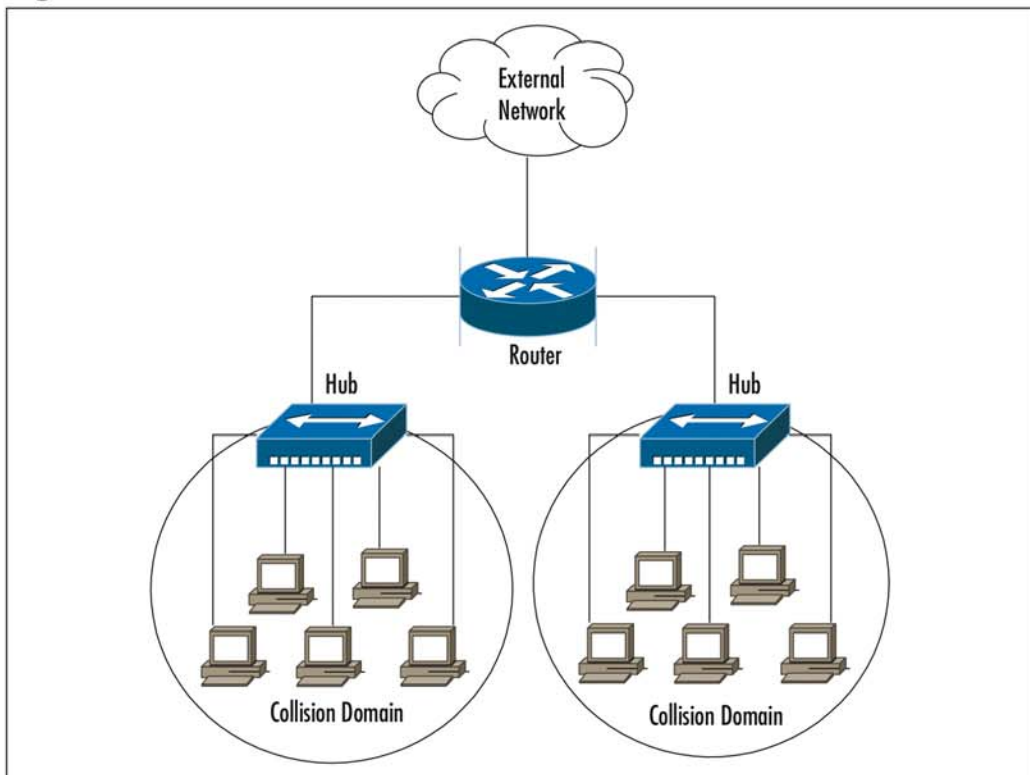
Hardware: Taps, Hubs, and Switches, Oh My!

Cable taps are hardware devices that assist in connecting to the network cable. Tap stands for Test Access Point, and you can use this device to access any cable between computers, hubs, switches, routers, and other devices. Taps are available in full or half-duplex for 10, 100, and 1000 Mbps Ethernet links. They are also available in various multi-port sizes. Following is a list of some popular cable tap products:

- Net Optics carries several types of network taps for copper and fiber cables. They can be viewed at www.netoptics.com.
- The Century Tap family is available by Shomiti at www.shomiti.net/shomiti/century-tap.html. They offer a variety of taps for copper and fiber cables.

A hub is a device that allows you to connect multiple hosts together on a shared medium, such as Ethernet. When a computer sends information, it travels into the hub and the hub blindly forwards the information to all other computers connected to it. As we explained before with Ethernet, the computer that the information was intended for will recognize its own MAC address in the packet header and then accept the data. The area that the hub forwards all information to is known as a *collision domain*, or *broadcast domain*. A hub has only one collision domain for all of the traffic to share. Figure A.4 shows a network architecture with collision domains related to hubs. Large collision domains not only makes sniffing easier, but also create performance issues like bandwidth hogging or excessive traffic on the hub.

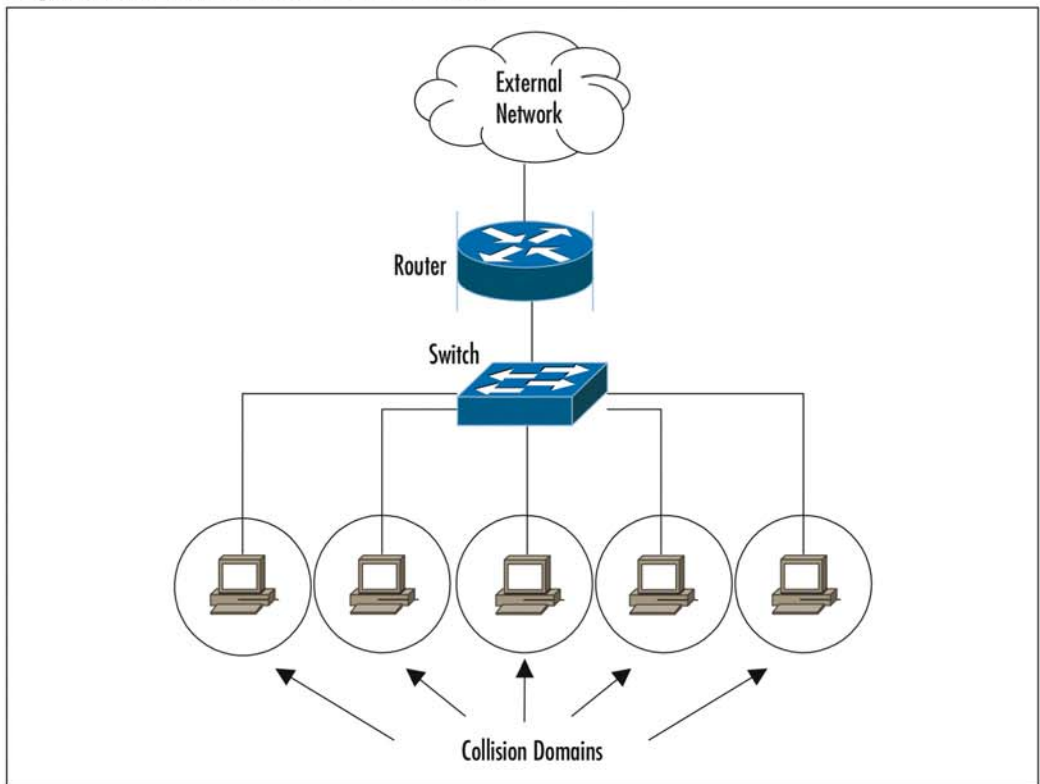
Figure A.4 Hub Collision Domains



A switch operates very differently from a hub. It is also used to connect computers together on a shared medium; however, when a switch receives information from a computer it doesn't just blindly send it to all other computers. A switch will

actually look at the packet header to locate the destination MAC address. A switch maintains a list of all MAC addresses and corresponding ports on the switch that the computers are connected to. It will then forward the packets to the specified port. This narrows the collision domain, or broadcast domain to a single port, as shown in Figure A.5. This type of collision domain will also provide a definite amount of bandwidth for each connection rather than a shared amount on a hub. Since the price of switches has fallen dramatically in the last few years, there is no reason to not replace hubs with switches, or to choose switches when purchasing new equipment. Also, some of the more costly switches often include better technology to make them more resistant to sniffing attacks.

Figure A.5 Switch Collision Domains



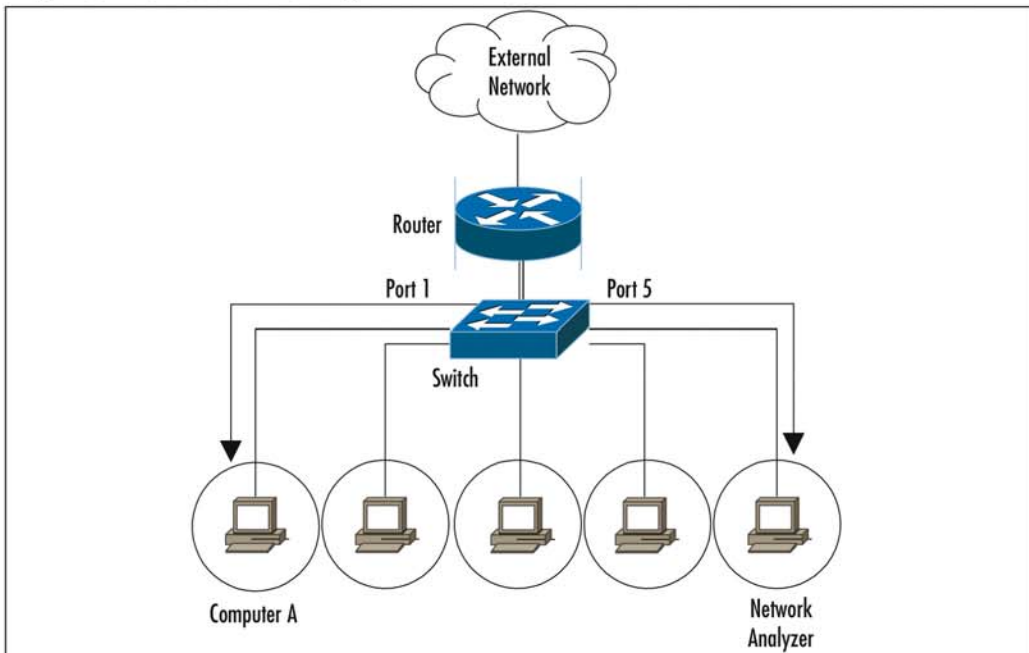
As you can see from the diagrams, hubs make sniffing easier, and switches make it more difficult. However, switches can be tricked, as discussed in the “Defeating Switches” section.

Port Mirroring

What if you are working in a network that uses switches and you want to perform network analysis legitimately? You are in luck, as most switches and routers come with a feature known as *port mirroring*, or *port spanning*. To mirror ports, you need to configure the switch to duplicate the traffic from a port you want to monitor to a port you are connected to with your network analyzer. This feature was designed just for this purpose, to analyze network traffic for troubleshooting.

Using port spanning does not interfere with the normal operation of switches, but you always want to check the documentation of the exact switch you are configuring and periodically check the device's logs. You won't affect the switch, but you will increase the amount of traffic on a specific destination port, so make sure your properly configured network analyzer is the destination port. Please consult the documentation for your specific switch to learn the exact command to enable port mirroring. Figure A.6 shows the process of port mirroring. The switch is configured to mirror all port 1 traffic to port 5. The network analyzer will see any traffic to and from Computer A. Sometimes administrators will mirror the uplink port on a switch; that way they will see all traffic to and from the switch and all of its ports.

Figure A.6 Port Mirroring



NOTE

Span means Switched Port ANalyzer. Cisco uses the word *span* to describe the concept of port mirroring. To span a port in Cisco terms is the same as mirroring a port.

Defeating Switches

We mentioned earlier that the use of switches in your network makes sniffing more difficult. In theory, on a switch you should only see traffic destined for your own computer. Notice we didn't say that switching eliminates sniffing. There are ways to trick a switch, or to get around its technology. The following list describes several ways in which a switch can be defeated:

- **Switch Flooding** Some switches can be made to act like a hub, where all packets are broadcast to all computers. This can be accomplished by overflowing the switch address table with all kinds of fake MAC addresses. This is known as a device *failing open*, thus removing all security provisions. Devices that *fail close* will incorporate some sort of security measure, such as shutting down all communications. The Dsniff package comes with a program called *macof* that is designed to perform switch MAC address flooding. It can be downloaded from <http://monkey.org/~dugsong/dsniff>.
- **ARP Redirects** When a computer needs to know the MAC address of another computer, it will send an ARP request. Each computer maintains an ARP table to store the MAC addresses of other computers that it has talked to. ARPs are broadcast on a switch, so all computers on that switch will see the request and the response. There are several methods that use ARP to trick a switch into sending traffic somewhere it shouldn't. First, an intruder can subvert a switch by sending out an ARP claiming to be someone else as the MAC address. An intruder can also send an ARP claiming to be the router, in which case computers will try to send their packets through the intruder's computer. Or, an intruder can send an ARP request just to one victim, claiming to be the router, at which point the victim will start forwarding packets to the

intruder. All of these tricks will allow an intruder to see information that he/she is not supposed to see.

- **ICMP Redirect** Sometimes computers are on the same physical segment, the same switch, but different logical segments. This means they are in different IP subnets. When Computer A wants to talk to Computer B it will send its request through a router. The router knows that they are on the same physical segment, so it will send an ICMP Redirect to Computer A letting it know that it can send its packets directly to Computer B. An intruder, Computer X, could send a fake ICMP redirect to Computer A, claiming that it should send Computer B's packets to Computer X.
- **ICMP Router Advertisements** These advertisements inform computers of who the router is. An intruder could send these types of advertisements out claiming to be the router, and computers will start to forward all packets through the intruder.
- **MAC Address Spoofing** An intruder can pretend to be using a different computer by spoofing its MAC address. Sending out packets with the source address of the victim will trick the switch. The switch will enter the spoofed information into its table and begin sending packets to the intruder. But what about the victim, who is still on the switch and sending updates causing the switch to change the table back? This can be solved by taking the victim offline with some sort of DoS attack, then redirecting the switch and continuing with communications. The intruder could also broadcast out the traffic that he receives to ensure that the victim computer still receives the packets. Some switches have a countermeasure that will allow you to statically assign a MAC address to a port. This may be difficult to manage if you have a large network, but it will eliminate MAC spoofing.

To spoof your MAC on Linux or Solaris when you are connected locally, you can simply use `ifconfig` as follows:

```
ifconfig eth0 down
ifconfig eth0 hw ether 00:02:b3:00:00:AA
ifconfig eth0 up
```

Register the MAC on all hosts by broadcast ping (and use Control C to close the ping): **ping -c 1 -b 192.168.1.255**

Now you can sniff all traffic to the computer that owns this MAC address.

- **Reconfigure port spanning on the switch** As we mentioned earlier, switch ports can be configured to see traffic destined for other ports. An intruder could perform this by connecting to the switch via Telnet or some other default backdoor. The intruder could also use SNMP if it is not secured.
- **Cable taps** As mentioned earlier, cable taps can be used to physically tap into the cable. Tapping into the uplink cable on a switch will show you all of the traffic entering and exiting that switch.

There are many methods of defeating switches, but this is contingent upon how a switch operates. Not all of the methods discussed will work, especially with newer, more technologically savvy switches. The Dsniff FAQ contains some good information for sniffing in a switched environment. It can be located at <http://monkey.org/~dugsong/dsniff/faq.html>.

Detecting Sniffers

Remember earlier that we said sniffers are a form of passive attack. They don't interact with any devices or transmit any information, thus making them very difficult to detect. Although tricky, detecting sniffers is possible. The easiest method is to check your network interfaces to see if they are in promiscuous mode. On UNIX-based systems the command **ifconfig -a** will list the network adapters on the system. Look for the PROMISC flag in the output, such as in the following example:

```
[root@localhost root]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:02:B3:06:5F:5A
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:204 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:46113 (45.0 Kb)  TX bytes:5836 (5.6 Kb)
          Interrupt:11 Base address:0x1800 Memory:e8120000-e8120038
```

If `ifconfig` is not detecting a sniffer that you know is currently installed and in promiscuous mode, you can try using the **ip link** command, a handy TCP/IP interface configuration and routing utility. The following example shows the output from the `ip` command:

```
[root@localhost root]# ip link
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:02:b3:06:5f:5a brd ff:ff:ff:ff:ff:ff
```

Detecting promiscuous mode on Windows systems is more difficult because there are no standard commands that will list that type of information. However, there is a free tool called `PromiscDetect`, developed by Arne Vidstrom, that will detect promiscuous mode network adapters for Windows NT, 2000, and XP. It can be downloaded from <http://ntsecurity.nu/toolbox/promiscdetect>. The following example shows the output of `PromiscDetect`, the D-link adapter is in normal operation mode, but the Intel adapter has `Ethereal` running on it:

```
C:\>promiscdetect
PromiscDetect 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
    - http://ntsecurity.nu/toolbox/promiscdetect/

Adapter name:
    - D-Link DWL-650 11Mbps WLAN Card
Active filter for the adapter:
    - Directed (capture packets directed to this computer)
    - Multicast (capture multicast packets for groups the computer is a member
of)
    - Broadcast (capture broadcast packets)
Adapter name:
    - Intel(R) PRO/100 SP Mobile Combo Adapter
Active filter for the adapter:
    - Directed (capture packets directed to this computer)
    - Multicast (capture multicast packets for groups the computer is a member
of)
    - Broadcast (capture broadcast packets)
    - Promiscuous (capture all packets on the network)
WARNING: Since this adapter is in promiscuous mode there could be a sniffer
        running on this computer!
```

Unfortunately some sniffers can cover their tracks by hiding the promiscuous flags. Also, if the sniffer was installed on a compromised system by using a rootkit, the intruder has most likely replaced commands like `ifconfig`. The following list describes several other methods that could be used to detect sniffers on the network:

- Monitor DNS reverse lookups. Some sniffers will perform DNS queries to resolve IP addresses to host names. Performing a network ping scan or pinging your entire network address space could trigger this activity.
- Send TCP/IP packets to all IP addresses on the same Ethernet segment, but with fake MAC addresses. Normally the network interface card will drop packets with the wrong MAC address. However, some systems, when in promiscuous mode, will answer with a reset packet (RST). This may also work in a switched environment since switches forward broadcast packets that they don't have MAC addresses listed for. Many newer sniffers have build in defenses for this technique by altering the way they handle MAC addresses.
- Carefully monitor hub ports. Ideally you would have a network diagram and your cables would be labeled. Then, if something unusual appeared, such as a new device or a newly active hub port, you would recognize it. However, in reality, wiring closets and cabling can be a nightmare. If your hubs are being monitored with a protocol such as SNMP via a network management system, you may be able to use this information to detect any unusual connects and disconnects.
- Remember how ARP is used to link IP addresses to MAC addresses. Normally an ARP is sent out as a broadcast to everyone. However, you could send out an ARP to a non-broadcast address, followed by a broadcast ping. No one should have your information in his or her ARP table except the sniffer because it was listening to all traffic, even the non-broadcast traffic. Therefore the computer with the sniffer would respond.
- Use a honeypot. A honeypot is a server that is set up to monitor the activity of intruders. It contains fake data and services. In this case you could create fake administrator or user accounts on the honeypot and then create connections across the network to it using clear text protocols such as Telnet or FTP. If there are sniffers monitoring for user

names and passwords they will see the honeypot and the intruder will eventually try to log into it. Honeypots run intrusion detection software to monitor activity, and special signatures can be added to trigger alerts when the fake accounts are used.

- **Carefully monitor your hosts.** This includes disk space, CPU utilization, and response times. Sniffers gradually consume disk space each day as they log traffic, and they can sometimes put a noticeable load on the CPU. When the infected computer's resources become consumed it will respond more slowly than normal.

There are several tools that can be used to detect sniffers on your network. Many of them are outdated and no longer actively maintained, and sometimes just hard to find. Also, newer sniffers have been rewritten to evade their detection. However, we want to take a moment to mention some of them.

- **PromiScan Ver 0.27** This is a free program by Security Friday that is up-to-date and actively maintained. It runs on Windows 2000 and XP and requires the WinPcap driver. It can scan the local network looking for remote promiscuous mode adapters, using ARP packets. It can be downloaded from www.securityfriday.com/ToolDownload/PromiScan/promiscan_doc.html.
- **AntiSniff** This program was originally written by L0pht, but is no longer supported or maintained. Archived Windows and UNIX versions can be downloaded from <http://packetstormsecurity.nl/sniffers/antisniff>.
- **Sentinel** This free program performs remote promiscuous detection, and runs on various versions of BSD and Linux. It requires the libpcap and libnet libraries to operate. It can be downloaded from www.packet-factory.net/projects/sentinel.
- **Neped** Network Promiscuous Ethernet Detector is a free UNIX-based program originally written by the Apostols Group to remotely detect promiscuous mode network interface cards on Linux computers. It only detects on a subset of Linux systems with unpatched kernels before version 2.0.36. The Apostols website no longer exists and neped can be difficult to find. Currently there is a version located at www.dsinet.org/tools/network-sniffers/neped.c.

- **Check Promiscuous Mode (CPM)** This is a free UNIX-based program developed by CERT/CC in response to increased network sniffing. More information, including the program, can be obtained from www.cert.org/advisories/CA-1994-01.html.
- **Ifstatus** This is a free UNIX-based program to detect promiscuous mode interfaces on Solaris and AIX systems. It can be downloaded from <ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/ifstatus>.
- **Promisc.c** This is a free UNIX-based program to detect promiscuous mode interfaces on Linux and some SunOS systems. It can be downloaded from www.dsinet.org/tools/network-sniffers/promisc.c.

Protecting Against Sniffers

So far you have learned what sniffing is and how it works. You have also learned some of the tricks that can be used by intruders to sniff where they aren't supposed to, and some not-so-foolproof methods of detecting sniffers. None of this sheds much of a positive light on your plight to protect your network and data. Fortunately there are some methods that you can use on your network that offer protection against the passive attack known as sniffing.

We talked earlier about using switches on your network instead of hubs. However, we also learned the methods used to defeat switches. Using switches is a network best practice that will allow increased performance and security that should be used regardless of existing methods to evade them. While switches will present a barrier to casual sniffing, the best method of protecting your data is encryption. Encryption is the best form of protection against traffic interception, on public networks as well as your own internal networks. Intruders will still be able to sniff the traffic, but the data will appear unreadable. Only the intended recipient should be able to decrypt and read the data. Some methods of encryption still leave the headers in cleartext, so the intruder will be able to see the source and destination addresses and possibly map the network, but the data will be obscured. Other forms of encryption will also mask the header portion of the packet.

A virtual private network (VPN) uses encryption and authentication to provide secure communications over an otherwise insecure network. VPNs protect the transmission of data over the Internet, and even your internal network. However, if an intruder compromises either of the end nodes of a VPN, the protection is rendered useless. The following list describes some of the VPN methods in use today that will protect your data against sniffing:

- **Secure Shell (SSH)** SSH is an application-level VPN that runs over TCP to secure client-to-server transactions. This is often used for general logins and to administer servers remotely. It is typically used to replace Telnet, FTP, and Berkley Services “r” commands. However, since any arbitrary TCP protocol can be tunneled through an SSH connection, it can be used for numerous other applications. SSH provides authentication by RSA or DSA asymmetric key pairs. The headers in an SSH session are not encrypted, so an intruder will still be able to view the source and destination addresses.
- **Secure Sockets Layer (SSL)/Transport Layer Security (TLS)** SSL was originally developed by Netscape Communications to provide security and privacy to Internet sessions. It has been replaced by TLS as stated in RFC 2246. TLS provides security at the transport layer and overcomes some security issues of SSL. It is used to encapsulate the network traffic of higher-level applications such as LDAP, HTTP, FTP, NNTP, POP3, and IMAP. It provides authentication and integrity via digital certificates and digital signatures.
- **IP Security (IPSec)** IPSec is a network-level protocol that incorporates security into the IPv4 and IPv6 protocols directly at the packet level by extending the IP packet header. This allows the ability to encrypt any higher layer protocol. It is currently being incorporated into routing devices, firewalls, and clients for securing trusted networks to one another. IPSEC provides several means for authentication and encryption, supporting quite a few public key authentication ciphers and symmetric key encryption ciphers. It can operate in tunnel mode to provide a new IP header that will mask the original source and destination addresses.

One-time passwords (OTP) is another method to protect against sniffing. S/key, One-time Passwords In Everything (OPIE), and other one-time password techniques will protect against the collection and reuse of passwords. They operate by using a challenge-response method, and a different password is transmitted each time authentication is needed. The passwords that a sniffer collects will be useless since they are only used once. Smart cards are a popular method of implementing one-time passwords.

E-mail protection is a hot topic for both companies and individuals. Two methods of protecting e-mail, by encrypting it in transit and in storage, are Pretty

Good Privacy (PGP) and Secure Multipurpose Internet Mail Extensions (S/MIME). Each of these methods also provides authentication and integrity by the use of digital certificates and digital signatures.

Network Analysis and Policy

There is one very important topic that we would like to take time to address. Before cracking open your newly installed network analyzer at work, please read your company policy! A properly written and comprehensive “Appropriate Use” network policy will more than likely prohibit you from running network analyzers. Usually the only exception to this is if network analysis is in your job description. Also, just because you may provide security consulting services for company clients, does not mean that you can use your sniffer on the company network. However, if you are an administrator and are allowed to legitimately run a sniffer, you can use it to enforce your company’s security policy. If your security policy prohibits the use of file sharing applications such as KaZaA, Morpheus, or messaging services such as Internet Relay Chat (IRC) or Instant Messenger, you could use your sniffer to detect this type of activity.

Also, if you provide security services for clients, such as an ethical hacker who performs penetration testing, be sure that the use of a sniffer is included in your Rules of Engagement. Be very specific about how, where, and when it will be used. Also provide clauses, such as Non-Disclosure Agreements, that will exempt you from the liability of learning confidential information.

Another word of caution: many ISPs prohibit the use of sniffers in their “Appropriate Use” policy. If they discover that you are using one while attached to their network, they may disconnect your service. The best place to experiment with a sniffer is on your own home network that is not connected to the Internet. All you really need is two computers with a crossover cable between them. You can use one as a client, and install server services on the other, such as Telnet, FTP, Web, and mail. Install the sniffer on one or both computers and have fun!

NOTE

You can also download packet traces from numerous websites and read them with your network analyzer to get used to analyzing and interpreting packets. The HoneyNet Project at <http://project.honeynet.org> has monthly challenges and other data for analysis.

Summary

Network analysis is the key to maintaining an optimized network and detecting security issues. Proactive management can help find issues before they turn into serious problems and cause network downtime or compromise confidential data. In addition to identifying attacks and suspicious activity, you can use your network analyzer data to identify security vulnerabilities and weaknesses and enforce your company's security policy. Sniffer logs can be correlated with IDS, firewall, and router logs to provide evidence for forensics and incident handling. A network analyzer allows you to capture data from the network, packet by packet, decode the information, and view it in an easy to understand format. Network analyzers are easy to find, often free, and easy to use; they are a key part of any administrator's toolbox.

We covered the basics of networking, Ethernet, the OSI model, and hardware that is used in a network architecture. Believe me, we only scratched the surface here. A good networking and protocols reference should be on every administrator's bookshelf. This will come in very handy when you discover some unknown or unusual traffic on your network.

As an administrator, you should also know how to detect the use of sniffers by intruders. You should keep up to date on the methods that intruders use to get around security measures that are meant to protect against sniffing. As always, you will also need to make sure that your computer systems are up to date with patches and security fixes to protect against rootkits and other backdoors.

We also covered a variety of methods used to protect your data from eavesdropping by sniffers. You should always remain up to date on the latest security technologies, encryption algorithms, and authentication processes. Intruders are constantly finding ways to defeat current security practices, thus more powerful methods are developed. A good example is the cracking of the DES encryption scheme and its subsequent replacement with Triple Data Encryption Standard (3DES).

Finally, remember the rule of network analysis—only do it if you have permission. A happy, curious, up-and-coming administrator could easily be mistaken as an intruder. Make sure you have permission or use your own private network to experiment.

Solutions Fast Track

What is Network Analysis and Sniffing?

- ☑ Network analysis is capturing and decoding network data.
- ☑ Network analyzers can be hardware or software, and are available both free and commercially.
- ☑ Network analyzer interfaces usually have three panes: summary, detail, and data.
- ☑ The five parts of a network analyzer are: hardware, capture driver, buffer, real-time analysis, and decode.

Who Uses Network Analysis?

- ☑ Administrators use network analysis for troubleshooting network problems, analyzing the performance of a network, and intrusion detection.
- ☑ When intruders use sniffers, it considered is a passive attack.
- ☑ Intruders use sniffers mostly to capture user names and passwords, collect confidential data, and map the network.
- ☑ Sniffers are a common component of a rootkit.
- ☑ Intruders are using sniffers to control backdoor programs.

How Does it Work?

- ☑ Ethernet is a shared medium that uses MAC, or hardware, addresses.
- ☑ The OSI model has seven layers and represents a standard for network communication.
- ☑ Hubs send out information to all hosts on the segment, creating a shared collision domain.
- ☑ Switches have one collision domain per port and keep an address table of the MAC addresses that are associated with each port.

- ☑ Port mirroring is a feature that allows you to sniff on switches.
- ☑ Switches make sniffing more difficult, however the security measures in switch architectures can be overcome by a number of methods, thus allowing the sniffing of traffic designated for other computers.

Detecting Sniffers

- ☑ Sometimes sniffers can be detected on local systems by looking for the promiscuous mode flag.
- ☑ There are several tools available that attempt to detect promiscuous mode by using various methods.
- ☑ Carefully monitoring your hosts, hub and switch ports, and DNS reverse lookups can assist in detecting sniffers.
- ☑ Honeypots are a good method to detect intruders on your network who are attempting to use compromised passwords.
- ☑ Newer sniffers are smart enough to hide themselves from traditional detection techniques.

Protecting Against Sniffers

- ☑ Switches offer some, but little protection against sniffers.
- ☑ Encryption is the best method of protecting your data from sniffers.
- ☑ SSH, SSL/TLS, and IPSEC are all forms of VPNs that operate at various layers of the OSI model.
- ☑ IPSec tunnel mode can protect the source and destination addresses in the IP header by appending a new header.

Network Analysis and Policy

- ☑ Make sure you have permission to use a sniffer on a network that is not your own.
- ☑ Read the appropriate use policies of your ISPs before using a sniffer.

- ☑ If you are hired to assess a computer network, and plan to use a sniffer, make sure you have some sort of non-disclosure agreements in place, because you may have access to confidential data.
- ☑ One-time passwords render compromised passwords useless.
- ☑ E-mail should be protected while in transit and storage with some type of data encryption method.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this appendix and to assist you with real-life implementation of these concepts. To have your questions about this appendix answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: I ran a switch flooding program against my switch and it didn’t do anything, why not?

A: Some newer switches are resilient to some of the older flooding tools.

Q: I have hubs daisy-chained through the floors of my company’s building, is that all one collision domain?

A: Yes! Hubs do not have any intelligence built into them to know where to send data, so they will blindly forward it on to everyone. So every hub that is connected together is seeing traffic for all ports.

Q: When I run Ethereal on my Linux system, I don’t see the PROMISC flag in the ifconfig -a output.

A: Ethereal uses the libpcap program to perform packet capturing and filtering. Some newer versions of libpcap use a different method of putting an interface into promiscuous mode that ifconfig cannot detect.

Q: Will adding encryption to my network decrease performance?

A: Yes, encrypting and decrypting data can be resource-intensive, depending on several factors including the type of encryption algorithm and length of the key. However, depending on your network architecture, end users may not notice the difference in performance.

Q: What if an attacker compromises a host that I am using a VPN client on?

A: Your VPN would basically offer a safe and secure environment for the attacker to run wild! For example, you connect your work laptop at home to the Internet over dial-up or high-speed Internet, and your system is compromised via a trojan. Your connections back to the office are secured via a VPN connection which gets enabled once you connect to your mail server or other protected work resources. The attacker then has the ability to access these resources that are otherwise protected by your VPN.

Q: I still don't understand how one-time passwords work.

A: Let me give you an example. You are provided with an RSA Secure ID hardware token. This is a small device that has a screen on it with some numbers that change every sixty seconds. These numbers are your responses to the RSA server challenges, i.e. your password. The token and the server are synchronized, so when you log in, the server presents you with a challenge, i.e. asks you your password, and you type in whatever number is showing on your Secure ID token screen at the time. You will be authenticated for this session, but next time you login it will be a different number, hence a one-time password.

Introducing Intrusion Detection Systems and Snort

Solutions in this Appendix:

- Introducing Intrusion Detection Systems
 - Answering Common IDS Questions
 - Fitting Snort into Your Security Architecture
 - Determining Your IDS Design and Configuration
 - Defining IDS Terminology
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introducing Intrusion Detection Systems

It's three o'clock in the morning, and Andy Attacker is hard at work. With the results from the latest round of portscans at hand, Andy targets the servers that appear vulnerable. Service by service, Andy fires off exploits, attempting to overflow buffers and overwrite pointers, aiming at taking over other peoples' servers. Some of these attempts are successful. Encouraged, Andy quickly installs rootkits on the compromised machines, opening backdoor access mechanisms, securing the machines enough to lock other attackers out, and consolidating control. Once that is accomplished, Andy begins the next round of scan-and-exploit, from the newly compromised machines.

It's three o'clock in the morning, and a shrill insistent beeping rouses Jennifer Sysadmin from her bed. Blearily, she finds her pager on the nightstand and stares at the message it displays. A customized message alerts her to a Secure Shell overflow attempt... outbound from one of her servers. She is startled into wakefulness. Throwing back the covers and grumbling about the tendency of network malefactors to attack during prime sleeping hours, she grabs her cell phone and heads purposefully for the nearest computer.

It's three o'clock in the morning, and across town, Bob Sysadmin is sleeping peacefully. No pager or cell phone disturbs his rest.

Is Bob's security that much better than Jennifer's, so that he can sleep soundly while she cusses and does damage control? Or has he also been compromised and just doesn't know it yet? With only this information, we don't know. And if he doesn't have an Intrusion Detection System (IDS), neither does Bob. IDSs are a weapon in the arsenal of system administrators, network administrators, and security professionals, allowing real-time reporting of suspicious and malicious system and network activity. While they are not perfect and will not show you every possible attack, IDSs can provide much-needed intelligence about what's really going on on your hosts and your network.

What Is an Intrusion?

To understand what "intrusion detection" does, it is first necessary to understand what an intrusion is. Webster's dictionary defines an intrusion as "the act of thrusting in, or of entering into a place or state without invitation, right, or welcome." For our purposes, an intrusion is simply unauthorized system or network activity on one of your computers or networks. This can take the form of a legitimate user of a system trying to escalate his privileges and gain greater access to

the system than he has been allowed, a remote and unauthenticated user trying to compromise a running service in order to create an account on a system, a virus running rampant through your e-mail system, or many other similar scenarios. Intrusions can come from the deliberately malicious Andy Attackers of the world, or from the terribly clueless Archibald Endusers of the world, who will click on every e-mail attachment sent to them, despite repeated admonitions not to do so. Intrusions can come from a total stranger three continents away, from a disgruntled ex-employee across town, or from your own trusted staff.

OINK!

Detecting malicious activity when it comes from your own employees or users is one of the most important purposes for IDSs in many environments. In fact, a properly implemented IDS that is watched by someone besides your system administrators may be one of the few methods that can actually catch a system administrator when she is doing something malicious. This is one of the main reasons why you should have network security personnel analyzing IDS events and system administrators managing systems.

Legal Definitions

Legally, there are not clear and universal standards for what constitutes an intrusion. There are federal laws about computer crime in many countries, such as the United States and Australia, but none in others. There are various state laws, and regional statutes in place, but not everywhere. Jurisdiction for computer crime cases can be unclear, especially when the laws of the attacker's location are vastly different from the laws in place in the compromised machine's region. To add to this confusion even if an intrusion is clearly within the legal definitions, many law enforcement agencies will not spend time working on it unless there is a clear dollar cost that is greater than some fixed amount. Some agencies use US\$10,000 for their guideline, while others use US\$100,000—this number varies from place to place.

Another legal concern when using IDSs is privacy. Technically, an IDS is a full content wiretap. In the United States, full content wiretaps are regulated by federal laws, including Title III of the Omnibus Crime Control and Safe Streets Act of 1968 (Title III), 18 U.S.C. §§ 2510–2522 and the Electronic

Communications Privacy Act of 1986. They are also subject to less stringent laws governing Pen Registers or Trap and Trace situations, such as the Pen Register, Trap and Trace Statute “Provider Exception,” 18 U.S.C. § 2511(2)(h). These generally involve tapping the characteristics and patterns of traffic without examining the data payload. Under these laws, intercepting network data may be illegal, particularly if it is not done by the network operator in the pursuit of his normal duties or in direct support of an ongoing criminal investigation of a computer trespasser. We strongly advise that you consult your legal department about your particular jurisdiction’s laws and the ramifications of deploying an IDS on your network.

Some enterprises rely on the status of their data as “protected trade secrets” under local common uniform trade secrets statutes. Such laws usually require the data to not be known to the public at large, and for some efforts to have been made to secure the data. Therefore, if you’re relying on such laws to save you when your data is stolen, you may be in for a nasty shock if the court deems your security measures insufficient. However, the U.S. Economic Espionage Act of 1996 (viewable at www.cybercrime.gov/eea.html) can make such activity a federal crime.

The type and scope of the activity can affect this as well. In computer security forums, there are often arguments about whether portscanning is legal. The answer depends on your jurisdiction. In 1998, Norway ruled that portscanning was not illegal. Michigan law, however, states that unauthorized use or access of a computer is illegal unless you have reason to think the system is designed for public access. Lawyers are still arguing about whether portscanning is “unauthorized use.” In some jurisdictions, login banners explicitly prohibiting access are required to prove that a given use of the system was unauthorized. Privacy expectations can play into the equation, too—if the user has an expectation that her system activity may be private, logging and prosecuting her for that activity may be difficult even if it is obviously malicious.

The best practices solution to this legal morass is usually to secure your systems as much as possible, clearly label all accessible services with login banners stating the terms of use, and know your local and federal computer crime laws, if there are any. That will help you protect your systems and identify what is considered an intrusion in your jurisdiction.

Scanning vs. Compromise

When watching network activity, one of the first things that usually jumps out is scanning activity; specifically, lots of scanning activity. Whether it's scanning for particular vulnerabilities or just scanning for open ports, this type of activity is very common on the unfiltered Internet, and on many private networks. Many IDSs are configured to flag scanning activity, and it's not uncommon to see the bulk of your alerts be caused by some form of scanning. While scanning is not necessarily malicious activity in and of itself, and may have legitimate causes (a local system administrator checking his own network for vulnerabilities prior to patching, for example, or a third-party company hired to perform a security audit of your systems), very often scanning is the prelude to an attempted attack. As such, many administrators want to be alerted when they are being scanned. Tracking scanning activity can also be useful for correlation in case of later attack.

Many popular network scanning tools are free, and freely available. A quick Google search will turn up everything from the ping and File Transfer Protocol (FTP) "Grim's Ping" to the full-featured portscanner Nmap, from the commercially available SolarWinds scanner to the vulnerability scanner Nessus. Since scanning tools are so easily accessible, it's not that surprising that they are so widely used.

However, it is important to realize that scanning is not an attempted compromise in and of itself, and should not be treated with the same level of escalated response that an actual attempted attack would merit. There are people who just scan systems out of curiosity and do not intend to attack them. A fellow that we met at a security conference once confided that before he engages in online financial transactions with any business, he scans all the company's machines that he can find. That's his way of determining whether he feels he trusts their security enough to trust them with his money.

It's also important to note that scanning activity is nearly constant. On the Wild West of the modern Internet, all sorts of automated programs are scanning large ranges of addresses, all the time. Some of them might be yours. Network monitoring tools, worms and viruses, automated optimization applications, script kiddies, and more are constantly probing your machines and your network. If you don't make a deliberate effort to filter it out, seeing this traffic on the Internet is a fact of life.

OINK!

While it is important to know when your network is being scanned, you don't want to make the mistake of spending your valuable time tracking down every fool who appears to be scanning your network. One of the best things you can do with information about scans is to track the source IPs that are scanning you and then use them to correlate against alerts for higher priority events or look for repeat scanners.

Viruses and Worms—SQL Slammer

Now that we've discussed scanning activity, let's get into a little more detail about some of the actual attempted compromises out there. Another very common type of traffic that you'll see triggering your IDSs is automated worms. Worms and viruses are often good candidates for IDSs, because they have repeatable and consistently identifiable behavior. Even polymorphic worms and viruses that attempt many attack vectors will have some network behavior in common, some traffic pattern that can be matched and detected by your IDS. As an example, let's look at the SQL Slammer worm.

On January 25, 2003, the SQL Slammer worm was released into the wild. Also known as Sapphire, the worm exploits a weakness in the Microsoft Structured Query Language (SQL) server. It sends a 376-byte User Datagram Protocol (UDP) packet to port 1434, overflows a buffer on the SQL server, and gains SYSTEM privileges, the highest possible level of compromise on a Windows operating system. Once it has successfully compromised a host, it starts scanning other IP addresses to further spread.

OINK!

Worms that use multiple attack paths are an excellent example of the value of correlation. The individual alerts from CodeRed or Nimda are common enough, but when they are seen together (as they would be from CodeRed or Nimda), they are a very distinct fingerprint for that worm.

It is also worth noting that SQL Slammer is a perfect example of a situation where an "active response" IDS would not be able to prevent infection, but an inline IDS would.

From the moment of its release, it is estimated that the worm spread world-wide in approximately 10 minutes. Massive amounts of network bandwidth were chewed up by the worm's scanning and propagation attempts. Many systems were compromised. Five of the 13 root Domain Name servers that provide name service to the Internet were knocked down by the worm. You can read the Microsoft advisory about the worm at www.microsoft.com/technet/treeview/default.asp?url=/technet/security/alerts/slammer.asp, and the Computer Emergency Response Team Coordination Center's (CERT-CC) advisory about the worm at www.cert.org/advisories/CA-2003-04.html.

OINK!

The CERT/CC is a center of Internet security expertise located at the Software Engineering Institute, a federally funded research and development center operated by Carnegie-Mellon University.

So, what's a good candidate rule for catching this with an IDS? Obviously, this is just the type of activity that you want to detect on your network. One thing common among every Slammer-infected host is the exploit payload it sends out. And indeed, that's exactly what the Snort IDS signature for the rule matches against. Here's the Snort signature that matches this activity:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL Worm propagation attempt"; content:"|04|"; depth:1; content:"|81 F1 03 01 04 9B 81 F1 01|"; content:"sock"; content:"send"; reference:bugtraq,5310; classtype:misc-attack; reference:bugtraq,5311; reference:url,vil.nai.com/vil/content/v_99992.htm; sid:2003; rev:2;)
```

You can see that the alert is labeled as an attempt at worm propagation, and that it matches UDP traffic headed to our network \$HOME_NET on port 1434 with a specific payload. Using this signature, we can detect and enumerate how many attack attempts we saw, and what hosts on our network they were attempting to reach. Massive automated attacks like this one usually engender a

coordinated response from the security community—IDS programmers writing new signatures, antivirus vendors writing checks and fixes, backbone providers tracking the traffic and mitigating its effect by filtering as requested and as needed. This signature can help us track infection attempts by the worm on our network, and make sure that our systems under attack remain secure.

Coordinating responses between companies and defenders is one of the few ways we can keep up with the attackers. A large number of organizations are dedicated to helping responders deal with attacks and share information.

OINK!

Here are some of the many organizations chartered to help mitigate attacks:

- The Forum of Incident Response and Security Teams, also known as FIRST, is a cluster of security professionals at various organizations. Membership is restricted to eligible teams with a clear charter and organizational scope, sponsored by an existing team, and capable of conducting secure communications with PGP.
- Information Sharing and Analysis Centers, or ISACs, were chartered in the United States in 1998 under the PDD 63, Protecting America's Critical Infrastructure policy. ISACs cover areas as diverse as electricity, financial services, drinking water, and surface transportation, but the most relevant ISAC for network security is the Information Technology ISAC, online at www.it-isac.org/.
- The Distributed Intrusion Detection System Dshield correlates firewall logs and reports of network attacks worldwide. Anyone can join, or submit his or her logfiles for analysis anonymously. Membership is free.
- Many commercial offerings will outsource your network security, firewall and IDS administration, log analysis, and attack correlation for you. Some providers will correlate data between their customers to increase the likelihood of detecting loud and active attackers, others will not. Specifics of the offered services depend on the vendor.

Live Attacks—Sendmail Buffer Overflow

We have seen what an IDS can do to let you know about an automated attack. However, what about attacks that are driven by a person, one single attempt at overflowing a network service rather than a virtual flood of packets? Snort can help with that, too. Let's look at an exploitable vulnerability, the Wingate POP3 buffer overflow.

The vulnerability is a remotely exploitable buffer overflow in the Wingate implementation of the POP3 daemon. After the USER command is sent, a sufficiently large amount of data following "USER" will overrun the buffer and may possibly lead to executing whatever exploit code is inserted. Normal use of the POP3 daemon would just supply a username after the USER command, and a normal username is unlikely to be very long. Now, let's look at the Snort rule that detects this attempted exploit:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 110 (msg:"POP3 USER overflow
attempt"; flow:to_server,established; content:"USER"; nocase;
isdataat:50,relative; pcre:"/^USER\s{^\\n}{50,}/smi"; reference:bugtraq,789;
reference:cve,CVE-1999-0494; reference:nessus,10311; classtype:attempted-
admin; sid:1866; rev:7;)
```

This rule looks for data with the content USER followed by more than 50 bytes of data, where those 50 bytes of data after USER don't contain a newline character. This should match the pattern of data we'd see in a real attempt at overflowing this buffer, and should not match legitimate user logins.

How an IDS Works

Now that we have looked at some of the capabilities of an IDS as far as alerting on malicious traffic, it's time to take a closer look at what exactly IDSs can keep an eye on, what data sources they use to do this monitoring, how they separate attack traffic from normal traffic, and some possible responses to seeing malicious traffic.

What the IDS Is Watching

Let's start by looking at what your IDS is able to see. This is going to depend greatly on what type of IDS it is, and where it's placed in your network. IDSs are classified by their functionality, loosely grouped into the following three categories:

- Network-Based Intrusion Detection System (NIDS)
- Host-Based Intrusion Detection System (HIDS)
- Distributed Intrusion Detection System (DIDS)

Network IDS

The NIDS derives its name from the fact that it monitors an entire network segment, or subnet. This is done by changing the mode on the NIDS' network interface card (NIC). Normally, a NIC operates in nonpromiscuous mode, listening only for packets destined for its own media access control (MAC) address. Other packets are not forwarded up the stack for analysis; they are ignored. To monitor all traffic on the subnet, not just those packets addressed to the NIDS machine itself, the NIDS must accept all packets and forward them up the stack. This is known as promiscuous mode.

In promiscuous mode, the NIDS can eavesdrop on all communications on the network segment. However, that's not all that is necessary to ensure that your NIDS is capable of listening to all traffic on the subnet. The network device immediately upstream of your NIDS must also be configured to send all packets on the subnet to your NIDS. If that device is a hub, all packets are automatically sent since all ports on a hub receive all traffic flowing through the hub. However, if that device is a switch, you may have to put the port on the switch into a monitoring mode, turning it into a span port. After setting up your NIDS, it is advisable to run a sniffing tool on the interface, to ensure that you can see all traffic on the subnet.

The advantage of a NIDS is that it has no impact on the systems or networks it is monitoring. It doesn't add any load to the hosts, and an attacker who compromises one of the systems being watched can't touch the NIDS and may not even know it is there. One downside of the monitoring is maxing out your span ports that you are allotted on a given network, and maxing out the bandwidth on the span itself. If you have 20 100MB ports spanning to one port, you begin filling up backplane... once that 5GB or 11GB backplane is saturated, you are in a world of hurt.

Tools & Traps...

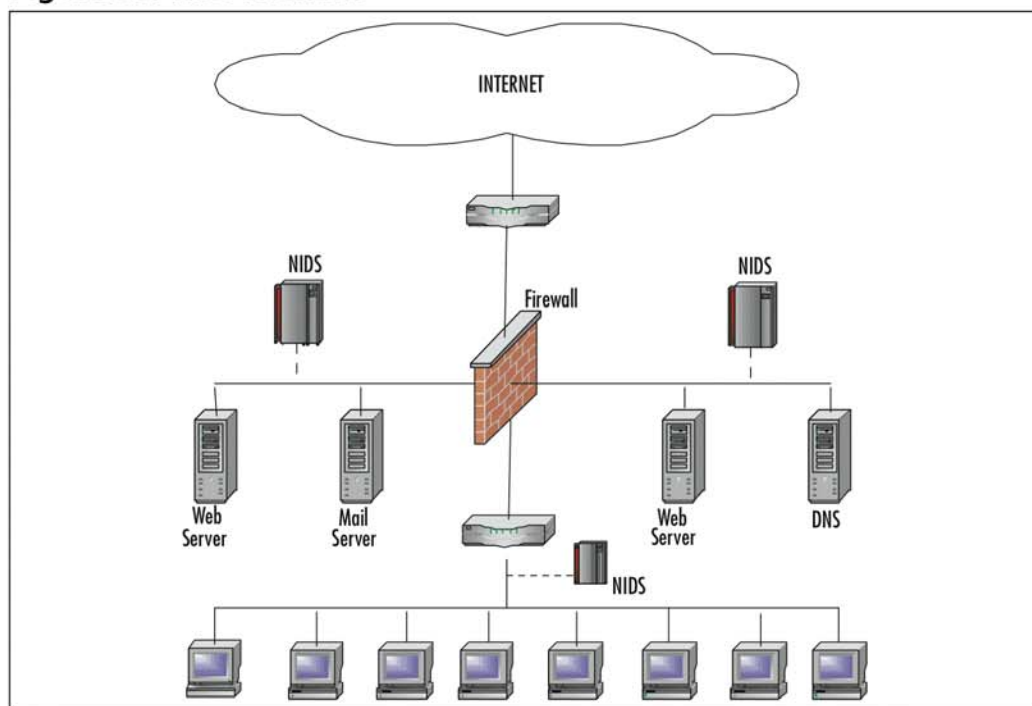
Network Sniffing Tools

When setting up or debugging a NIDS, it is vital to ensure that you are seeing all the traffic for the subnet to which you are connected. Snort is capable of functioning as a fine packet sniffer. When invoked from the command line with the `-i` switch, Snort will listen on a particular interface. Make sure you see traffic from and to other machines on the network, not just the broadcast traffic and the traffic to the local machine.

In addition to Snort, several other programs are perfectly good packet sniffers. Ethereal, available from www.ethereal.com, is a cross-platform packet sniffer. Tcpdump (www.tcpdump.com) is present on many Unix systems already, and Windump (<http://windump.polito.it>) serves the same function for Windows, although it usually will have to be installed on the system.

In view of emerging privacy regulations, monitoring network communications is a responsibility that must be considered carefully. Make sure that you are familiar with your local legal requirements for such activity.

In Figure B.1, we see a network using three NIDS. The units have been placed on strategic network segments and can monitor network traffic for all devices on the segment. This configuration represents a standard perimeter security network topology where the screened subnets housing the public servers are protected by NIDSs. When a public server is compromised on a screened subnet, the server can become a launching platform for additional exploits. Careful monitoring is necessary to prevent further damage.

Figure B.1 NIDS Network

The internal host systems are protected by an additional NIDS to mitigate exposure to internal compromise. The use of multiple NIDS within a network is an example of a defense-in-depth security architecture.

OINK!

In case you missed it, let's say that again—privacy regulations can be a dangerous trap. Even if you have your users sign an Acceptable Use Policy that stipulates you have the right to watch them, there may still be situations where they can claim an assumption of privacy. Be sure to get approval from your management (if you are the one deploying the IDS), or your Human Resources department (if your company has one), or as a last resort, talk to your lawyer and make sure you aren't violating any laws. If you do this incorrectly, you may find that *you* are being prosecuted instead of the person you were trying to monitor! The PATRIOT Act, despite its many critics, does appear to grant the service provider and system administrators the ability to monitor the use of their networks and systems for the purpose of identifying misuse.

Careful consideration must be paid to who sees the data, and to the process of keeping that data secure. Finally, remember that any legal advice given in this book is not offered by a lawyer—you should check it with your own before depending on it.

Host-Based IDS

Host-based IDSs, or HIDSs, differ from NIDSs in two ways. First, an installed HIDS protects only the system on which it resides, not the entire subnet, and second, the network card of a system with a HIDS installed normally operates in nonpromiscuous mode. This can be an advantage in some cases—not all NICs are capable of promiscuous mode, although most modern NICs are. In addition, promiscuous mode can be CPU intensive for a slow host machine.

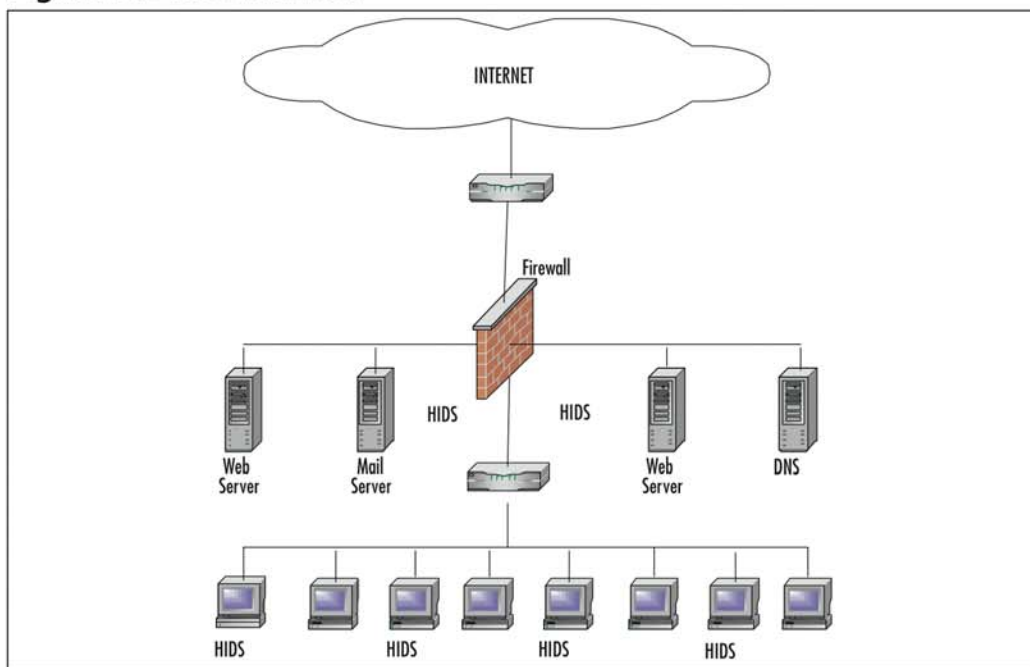
Another advantage of HIDS is the ability to tailor the ruleset to be very specific to the particular host system. For example, there is no need to configure multiple rules designed to detect Network File System (NFS) exploits on a host that is not using the NFS. Being able to fine-tune your ruleset will enhance performance and decrease false positives (or true positives that you simply don't care about). The major advantage of a HIDS, however, lies in its capability to detect specific changes to the files and operating system of its host. It can monitor file sizes and checksums to ensure that crucial system files are not maliciously modified without someone noticing. It can intercept rogue system calls that may be an attempt at exploiting a local vulnerability. Moreover, it can watch traffic within a system that never crosses the network, and therefore would never be seen by the NIDS.

There are a few downsides to electing to use a HIDS. You'll have to choose one that is tailored to your operating system. If you have many different operating systems on your network and want to use the same vendor for all your HIDSs, you may have to do a little shopping to find the right vendor that supports all of your operating systems. A HIDS will add load to the host on which it is configured, as the HIDS process(es) will consume resources. This is usually not a problem on an individual's desktop, but can become one on a busy network server. Make sure you are familiar with the details of any HIDS that you choose and how it operates—some HIDSs will watch file accesses, usage times, process loads, and/or system calls, while others may also watch network activity from the point of view of that host. Know what features you want in your HIDS, and make sure that the HIDS you select will support those features on all the platforms you need.

In addition, maintaining a large network of systems with many HIDS deployed can be very challenging. The HIDS solution alone does not always scale well, and without centralized management, you may be a very busy system administrator indeed trying to keep up with all those alerts.

Figure B.2 depicts a network using a HIDS on specific servers and host computers. As previously mentioned, the ruleset for the HIDS on the mail server is customized to protect it from mail server exploits, while the Web server rules are tailored for Web exploits. During installation, individual host machines can be configured with a common set of rules. New rules can be loaded periodically to account for new vulnerabilities.

Figure B.2 HIDS Network



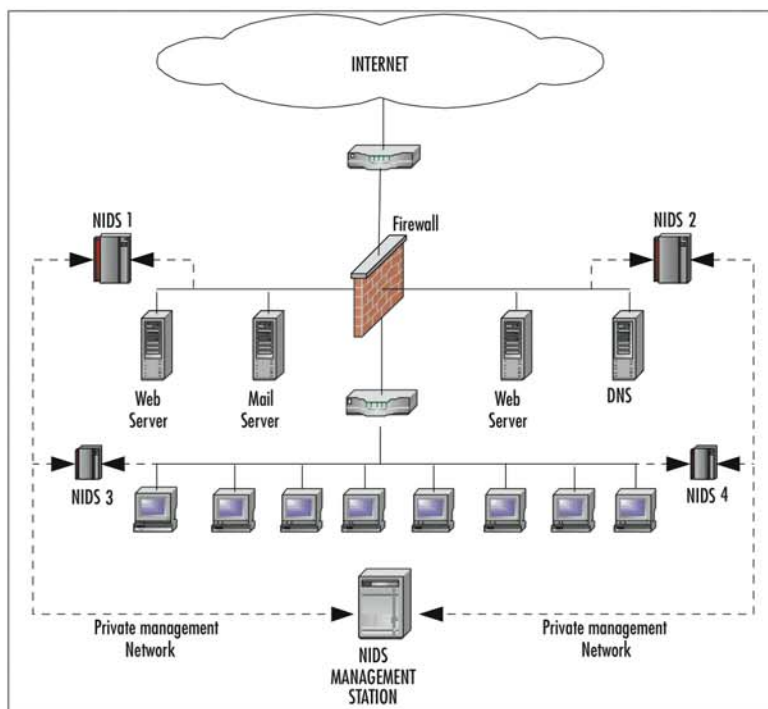
Distributed IDS

A Distributed Intrusion Detection System, or DIDS, is a combination of NIDS sensors, HIDS sensors, or both, distributed across your enterprise, and all reporting to a central correlation system. Attack logs are generated on the sensors and uploaded (either periodically or continuously) to the central server station where they can be stored in a central database. New attack signatures are created

or downloaded to the management station as they become available, and can then be downloaded to the sensors on an as-needed basis. The different kinds of sensors may or may not be managed by the same server, and the management servers are frequently separate from the servers that collect the logs. The rules for each sensor can be tailored to meet its individual needs, suiting the network or the host that each sensor monitors. Alerts can be forwarded to a messaging system located on the correlation system station and used to notify the IDS administrator.

In Figure B.3, we see a DIDS system comprised of four sensors and a centralized management station. Sensors NIDS 1 and NIDS 2 are operating in stealth promiscuous mode and are protecting the public servers. Sensors NIDS 3 and NIDS 4 are protecting the host systems in the trusted computing base.

Figure B.3 DIDS Network



The network transactions between sensor and manager can be on a private network, as depicted, or the network traffic can use the existing infrastructure. When using the existing network for management data, the additional security

afforded by encryption, or VPN technology, is highly recommended. Sending all the security information about your network across it in cleartext is just asking clever attackers to intercept those communications. At best, they can tell when they are triggering your IDS, and can tailor their behavior to avoid detection. At worst, they could intercept and change your alerting mechanism, hopelessly corrupting your data and any chance you might have of relying on it for analysis and/or prosecution. Another issue to keep in mind if you choose to have your DIDS communicate over your normal network is that if your company network is ever flooded or disabled by malicious traffic (as happened to many networks as a result of SQL Slammer), your IDS sensors won't be able to communicate with the correlation or management servers, which significantly reduces their usefulness.

OINK!

We'll refer to "stealth mode" for NIDS on occasion. This means that the NIDS is not visible to the network it is monitoring. This is generally done by not giving an IP address to the NIC that is being used for monitoring, and by using a device known as a "Tap" that only allows the receipt of traffic, not sending it. This method of watching network traffic is key to preventing attackers from knowing about your NIDS.

One of the main advantages of analyzing events using DIDs is to be able to observe system-wide, or even Internet-wide incidents from the 50,000-foot view. What might look like an isolated portscan to a class C subnet could look like a global worm propagating to a system like Dshield.

A friend of this book's editors, and frequent contributor to Dshield, is responsible for performing intrusion detection on two class Cs on opposite ends of a class B. He will watch a scan come through the lower class C, and return minutes later on the higher class C. DIDSs can be fairly complex to design, and require a talented hand to tune them and correlate and manage the data that is generated by all the sensors. The scope and functionality of the system varies greatly from implementation to implementation. The individual sensors can be NIDS, HIDS, or a combination thereof. The sensor can function in promiscuous mode or nonpromiscuous mode.

Now that we are familiar with how different types of IDSs can be deployed, let's look at the information they can gather.

Application-Specific Information

All three types of IDSs are able to watch at least some application-specific information. This can vary from the traffic that goes to and from your Web server to the internal data structures of your custom-coded application. (Of course, for a custom application, you'd have to have custom IDS rules to match its traffic.) As application traffic goes over the wire across your network, the NIDS will be able to detect it. If it's sent in cleartext like Telnet or HyperText Transfer Protocol (HTTP) traffic, the NIDS should have no problem matching against it. For example, look at this signature, looking for access to a vulnerable PHP: Hypertext Preprocessor (PHP) application "Proxy2.de Advanced Poll 2.0.2."

Tools & Traps...

PHP and Shifting Acronyms

At its inception, PHP stood for "Personal Home Page." It was, according to the PHP history at www.cknow.com/ckinfo/acro_p/php_1.shtml, a wrapper for Web pages around Perl. Over time, as the functionality of PHP shifted into a full-blown server-side scripting language for Web servers, the acronym came to mean nothing, and then to the current recursive acronym "PHP: Hypertext Preprocessor," as described at www.php.net/manual/en/faq.general.php.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-PHP  
Advanced Poll admin_tpl_misc_new.php access"; flow:to_server,established;  
uricontent:"/admin_tpl_misc_new.php"; nocase; reference:bugtraq,8890;  
classtype:web-application-activity; sid:2299; rev:2;)
```

Even if the traffic is sent in binary format, if there is a known payload or a consistent part of the packet (that is unique to avoid false positives) that the NIDS can match against, signature-based rule matching may be possible. Encrypted application traffic that is sent with sufficiently good cryptography, though, may be outside the scope of what most NIDSs are capable of detecting. Writing a good NIDS rule for traffic encrypted with a good random seed (for example, the same input string results in a different output every time it is encrypted) would be difficult.

Encrypted traffic is where host-based IDSs shine. Application traffic that crosses the network in an encrypted format is usually decrypted at each end-point. Consequently, traffic that was previously randomized gibberish becomes sensible patterns on the host, and can be matched against with signatures.

What types of things does one look for in application-specific information? Attempts to exploit input fields by entering too much data, known overflows or underflows exploiting lack of input validation, and attempted SQL injection are only a few possibilities. Of course, the signatures will vary greatly depending on the application that's being protected.

OINK!

Even though we just said that HIDSs shine when it comes to looking inside encrypted data because they are on the host that is sending or receiving the data (and as a result are more likely to see the information before it is encrypted or after it is decrypted), that isn't completely true. It is important to remember that they are only better if they are actually seeing the unencrypted data. That means that if the encryption is occurring at the application layer (for example, your Web browser or an SSH client or e-mail client is doing the encryption) and your HIDS is seeing the network traffic as it leaves or enters your system, after the encryption has taken place or before the decryption has taken place, then it doesn't matter that it is a HIDS; it still can't see through the encryption and is just as blind as any NIDS would be.

Host-Specific Information

While most HIDSs don't actually watch *everything* that happens on a host, they are capable of seeing all the behavior of a given host, from file creation and access to system calls to local network activity to the loopback interface. It is very common for HIDSs to create a database of the state of the system (file sizes, permissions, access times) when they are installed, and then monitor for deviations from that baseline. In fact, for many types of HIDSs the tuning process requires installing the HIDS software and then progressing with normal system activity to establish a baseline of what is changed when, and by whom.

Subnet-Specific Information

Most networks have common patterns to their traffic flows. If you know that one machine on your network is a mail server, you will not be surprised to see Simple Mail Transfer Protocol (SMTP) traffic going to and coming from it. If you are used to seeing a network-monitoring device ping every device on your network every five minutes, that traffic is acceptable even though the same behavior from another device on your network would be worrisome. Over time, a good NIDS should be tuned to recognize the expected behavior of the subnet on which it resides, permitting traffic that is known to be expected and acceptable, and sending alerts for similar traffic from unauthorized hosts. The workstation of your authorized pen-tester may scan your network, while the workstation of your new intern may not.

OINK!

The NIDS deployment described in the previous paragraph is frequently referred to as a policy-based IDS. It is most effective in environments where you have strict control over what type of traffic is acceptable. As a result, it is very common in military deployments or for companies that exercise extremely tight control over their networks and systems. If you have a very dynamic or extremely complex environment, it may be harder to implement a strict policy-based IDS approach.

Another worthwhile and often overlooked component of the subnet traffic is the Layer 2 protocol mapping that can be done. Most IDSs overlook Address Resolution Protocol (ARP) traffic, used to map MAC addresses to IP addresses on the local network. It is possible for attackers to spoof traffic by changing their MAC address or forging an IP address that is not theirs and then trying to intercept the return traffic. This type of tomfoolery may be viewable at the subnet level, depending on your network topology. If your NIDS is not on the same local subnet on which the Layer 2 attacks are happening, it will not detect them correctly. When network traffic crosses a router, the MAC address changes. Since we're checking for the ports and locations of MAC addresses, we cannot afford to have them change before examination or the data becomes unreliable. Therefore, if you want to capture Layer 2 data with your NIDS, ensure that your

NIDS is on the same local subnet as all the machines you want to monitor, before any routers become involved in the data stream.

Distributed IDS

All of the information can be collected and correlated with a DIDS, but the scale is much greater. Instead of getting the local-network view of your subnet and its machines, you get a view of the activity across your entire enterprise. You can pick out data patterns that would have been baffling or inconsequential at a smaller scale, and what seems to have been an automated backup of one server turns out to be a coordinated (malicious) replication of data network-wide, when you look at the big picture. Looking at traffic from the DIDS level allows you to see large-scale data flows and overall trends more clearly. The downside is that you must have the tools to effectively comprehend the amount of data you are collecting; otherwise, the subtle attacks that you had hoped to discover will be lost in the general noise from your environment.

How the IDS Watches Your Network

Without an effective method of collecting data to analyze, there really isn't any purpose to an IDS. Luckily, there are several possible ways for your IDS to collect data to analyze. The following are the most common methods of collecting data for your IDS to analyze. Each has its own strengths and weaknesses, and all are best suited for different tasks. There are several possible sources of data for your IDS.

Packet Sniffing

Any IDS that looks at network traffic performs packet sniffing. As we mentioned, NIDSs operate by setting an interface into promiscuous mode and packet-sniffing on that interface. By doing so, they capture each packet that crosses the wire on the local subnet. They will not see packets that cross a TCP/IP stack internal to a machine, but they will potentially see everything on the local wire. However, many HIDSs that perform analysis of network traffic also use similar techniques without the use of promiscuous mode, to collect traffic specific to the host on which they reside. Packet sniffing is a classic way of doing intrusion detection, and there are equally classic techniques of IDS evasion that can be used against packet sniffing IDS; for example, fragmentation attacks, which split the attack payload among several packets. We discuss evasion techniques and provide some key references later in this appendix. We strongly encourage you to read them and then keep them in mind when listening to vendors talk about

never missing an attack. The IDS response to this was to create the capability for the IDS to reassemble packets and then match against the assembled packet. The attacker response was to change the way the packets are fragmented, causing some data to overwrite itself. Then, IDS techniques were created for that, and so on, and so on. In case you hadn't guessed, Snort uses packet sniffing.

Log Parsing

Another excellent source of security data is from system log files. Many IDS systems can pull data from the system logs and alert if they see anything anomalous. In fact, some of the original IDS implementations used log monitoring as their data collection method. Some attacks are blatant in the footprints they leave in your system logs; the Secure Shell CRC32 overflow, for example, can leave

```
sshd[3698]: fatal: Local: crc32 compensation attack: network attack  
detected
```

in your logs.

INK!

Dr. Tina Bird has done quite a bit of work in log analysis of intrusion attempts; you can read the results of her research at www.loganalysis.org.

System Call Monitoring

HIDSs are capable of setting themselves up as resident in the operating system's kernel, and watching (or in some cases intercepting) potentially malicious system calls. A system call is a request that a program makes of the operating system kernel. If the HIDS thinks that the system call might be malicious, such as requesting a change of one's user ID to that of the root user, it can create an alert or, in the case of some HIDSs such as the Linux Intrusion Detection System (LIDS), disallow the system call unless specifically overridden.

Filesystem Watching

Another very common tactic of HIDSs is to keep an eye on the sizes and attributes of crucial files in a filesystem. If your operating system kernel suddenly

changes size and none of your system administrators knows anything about it, this is probably something to check into. If you find yourself with world-writable directories or you find that your common system binaries have changed, it's possible that they have been Trojaned. Watching the filesystem like this helps alert administrators to possible malicious activity; if not before the fact, at least as soon after as possible. Tripwire is perhaps the best-known example of a tool to monitor files for changes, but there are many others that do the same thing, including the open-source tool Advanced Intrusion Detection Environment (AIDE).

How the IDS Takes the Data It Gathers and Finds Intrusion Attempts

Any IDS is going to collect a vast amount of data—networks are busy, servers are buzzing, there is data transfer constantly going on, processes constantly being run, and a general low hum of electronic noise on your network. To be effective, an IDS must have at least one (and possibly several) algorithm for determining what traffic is worth the attention of your administrators. There are several strategies, but at the most basic level there are two tactical options.

Known Good versus Known Bad

Network traffic can be identified and classified in several fashions. You can seek to have your traffic conform to a given security policy, dictated by the particular needs of your enterprise or your network. Some administrators choose to only allow traffic that they know to be good, while others choose to only block traffic that they know to be bad. Most often, policy-based approaches will center on a known-good approach. To make the best decision for your enterprise, consider what types of traffic you are likely to see, how much staffing you have to deal with the alerts, and how paranoid you want to be.

Do you want to identify the known acceptable traffic on your network, and flag on everything else, or do you want to identify the known attacks and let everything else go by without comment? That's the basic conundrum of IDS strategy; firewall administrators are no doubt familiar with the dilemma. The known-good strategy will be orders of magnitude more work, as you try to sort through all the traffic on your network, determining what is supposed to be happening and what is dodgy. You'll immediately be faced with a large amount of false positives spewed forth by a frantically busy IDS, and will have to slowly winnow them down to a manageable level as you identify the known-good traffic on your network. In addition, unless nothing ever changes on your network, you will have

to constantly tune and retune the IDS to adjust to the normal changes that happen over time in almost any environment. There are automated tools for defining “normal,” where “normal” is expected to be an acceptable approximation of “good.” However, such tools suffer from issues of false positives in complex or highly dynamic environments. They can also be tricked into deciding that something is “normal” if the new activity occurs in small enough amounts over a long enough period of time. (Think of the story of boiling a frog—if you drop a frog in boiling water it jumps out. If you put a frog in cool water and slowly raise the temperature, it won’t notice and will simply be cooked.)

However, following a strategy of only alerting on known or suspected malicious traffic will result in much lower alert volume. In addition, because the rules can be very specific about what the definition is of something bad, when an alert does go off (assuming the rules are well written), you can be fairly confident that the “bad” activity was actually seen. This means that the person monitoring the IDS doesn’t have to be as skilled (because he doesn’t have to be able to troubleshoot the IDS), which can be a significant issue. However, this approach carries the strong likelihood of missing attack traffic that doesn’t happen to match your rules or algorithms, and if you write more flexible rules, the number of false positives will go up. In some scenarios, such as with Archibald Enduser’s home box, where Archibald doesn’t know a lot about intrusion detection and doesn’t have the time or inclination to learn, this may be the better solution. However, if you want to increase your likelihood of catching a given attack, and you have the resources available to monitor and maintain the IDS, you might want to consider the other approach. Your choice of strategy is a cost/benefit analysis; weigh the time and resources that you are willing to devote to IDSs with the importance of catching the maximum number of attacks.

OINK!

In reality, most well-planned IDS implementations use a combination of both approaches. Where you can tightly define allowed traffic, use a “known-good” approach. Where you have to be a little more permissive or the environment changes too frequently to define, use “known-bad.” Use each where it makes sense and you’ll be a much happier intrusion analyst.

Technologies for Implementing Your Strategy

IDSs differentiate attack traffic from innocuous network and system activity in several ways. Some primarily use a technique called *rule-based* (a.k.a *signature-based*) *analysis*, matching a known pattern to activity seen on the system or network. We have seen examples of Snort rules already, looking for packet content on the network and matching it to a series of predefined rules. The same thing can be done when looking at entries in log files or sets of system calls. This is very similar to the way many antivirus programs use virus signatures to recognize and block infected files, programs, or active Web content from entering a computer system (and why you have to constantly update your anti-virus software). Signature detection is the most widely used approach in commercial IDS technology today, since it is easily demonstrable, effective, and very customizable with limited training or experience. As new attacks are developed and seen in the wild, new signatures can be written to match and alert against the new forms of attack.

A more complex version of rule-based analysis is protocol analysis. Instead of writing a relatively simple rule that defines something about a specific event (good or bad), protocol analysis attempts to define every possible acceptable behavior for a specific kind of activity. For example, when our computer wants to set up a TCP connection, it sends a SYN packet. The acceptable responses are either RST/ACK or SYN/ACK. Anything else would be a violation of the protocol. This approach allows a little more flexibility in defining what “bad” is. Instead of saying, “If you see a string of greater than 500 bytes, filled with a specific character, it is an attack of this type,” you can say, “At this point in the connection, you should not see strings greater than 500 bytes. If you do, it is an attack. If you see more than 500 bytes at some other point in the connection, it is okay.” The problem is that while protocols are tightly and clearly defined, not all vendors choose to pay attention to everything in the protocol definition. As a result, you may find that your protocol analysis-based IDS is correctly complaining about something that is not allowed in the RFC (Request For Comments—the documents used to define most Internet protocols. For a full list, see www.rfc-editor.org) but is completely normal for applications from a specific vendor. In addition, it is tremendously time consuming and complex to write a good protocol model, and to implement it in an efficient enough fashion that it can be used to watch high-speed networks. This takes years of experience. This means that most vendors tend to be *very* unwilling to share their protocol models openly, even with customers. Consequently, troubleshooting false positives for protocol analysis IDS, or getting a false positive fixed can be a long process while you wait for your vendor. Another approach is called *anomaly*

detection. It uses learned or predefined concepts about “normal” and “abnormal” system activity (called *heuristics*) to distinguish anomalies from normal system behavior and to monitor, report on, or block anomalies as they occur. Some anomaly detection IDSs come with predefined standards for what normal network traffic should look like, and others watch the traffic on your network (or activities on your systems) and use a learning algorithm to develop a baseline profile from that. These profiles are baselines of normal activity and can be constructed using statistical sampling, a rule-based approach, or neural networks, to name just a few of the methods.

Literally hundreds of vendors offer various forms of commercial IDS implementations. Because of the simplicity of implementation, the majority of implementations are primarily signature based, with fewer protocol analysis solutions and only limited anomaly-based detection capabilities present in certain specific products or solutions.

OINK!

While most effective IDS deployments combine network- and host-based IDS implementations, very few vendors have been able to successfully offer both kinds of IDSs or IDSs that combine multiple technological approaches. The products end up doing everything in a barely acceptable fashion but nothing tremendously well. This may actually be changing due to the large number of acquisitions that we’ve seen in the IDS space in recent years. The vendors who are left may actually have the resources to dedicate to each separate area of focus, or they may just manage to do a miserable job in all the areas—which is what we’ve seen so often after acquisitions in the past.

What the IDS Does When It Finds an Attack Attempt

Most modern IDSs include some limited automatic response capabilities, but these usually concentrate on automated traffic filtering, blocking, or disconnects as a last resort. Although some systems claim to be able to launch counterstrikes against attacks, best practices indicate that automated identification and back-trace facilities are the most useful aspects (and the ones least likely to get you sued) that such facilities provide and are therefore those most likely to be used. There are different and highly configurable approaches to what the IDS actually

does when it detects an intrusion attempt. It is worth discussing briefly the merits of active IDS response (sometimes mistakenly known as IPS, or Intrusion Prevention Systems) versus the more traditional passive detection and alerting.

Passive Response

Traditionally, IDSs will watch the activity, and can be configured to log to a file and/or send alerts to the administrator(s). These alerts can take many forms—Simple Network Management Protocol (SNMP) traps, outgoing e-mails, pages or text messages to the system administrator, even automated phone calls. Most administrators configure the IDS to alert them in various ways depending on the severity of the perceived attack and the frequency of its occurrence. You don't want to be paged 10 times an hour for something that seems dire at first but turns out to be a false positive every time. However, you do want to be notified for an alert indicating a serious compromise, especially if it doesn't false-positive very often.

Traditional IDSs stop there. They are usually set up with a management interface entirely separate from their listening tap on the network, so that they don't betray their presence on the tap by sending alerts all the time. Very often, the listening tap doesn't even have an IP address, and is a stealth interface configured not to respond to any traffic.

Active Response

IDSs with Active response capabilities and IPSs emulate all the behavior of traditional passive IDSs as far as detection goes. However, when they see an attempted attack, they can be configured to take proactive measures against it rather than just alerting the administrator and waiting for him to take action. They can be placed inline and drop traffic they see as malicious, they can spoof Transmission Control Protocol (TCP) resets to either the source or destination systems (or both) to abruptly terminate a TCP session that they see attack traffic coming through, or they can send Internet Control Message Protocol (ICMP) Unreachable messages to the source system in an effort to convince it that the target system is unreachable; some reconfigure firewalls or routers between the targets and the attackers to block the traffic. Some systems will do nameserver lookups or traceroutes on the attacking system in an attempt to gather information about it. Some will even portscan the attacking system back, and give you a report of its likely operating system and possible vulnerabilities.

The appeal of active response is that you don't have to have a system administrator watching the wire in real time. The peril is that the consequences of a misconfiguration become much graver. We have set up brand new IDSs with prevention capabilities before, only to watch them listen to the network traffic, decide that our DNS server was portscanning the network, and block all access to it. Without name service, many network applications come to a screeching halt. IPSs should be checked for whitelisting capabilities beforehand in order to avoid just such scenarios. It would also be advisable to check the legalities in your jurisdiction if you're planning to have your system automatically trace or scan "attacking" systems.

Inline IDS

Another common configuration debate is whether your IDS should sit on a tap on your switched network, or sit inline between you and the Internet. There are advantages and disadvantages to both configurations. If you intend to have your IDS act as an IPS, setting it inline might be something you would strongly want to consider. Prevention is far more effective when the IDS is capable of simply dropping traffic that it has determined should not be allowed through. When your IDS is not inline, you can send ICMP unreachable or TCP Resets to both source and destination, but you have to hope that the devices themselves behave properly. You're not controlling the network segment between them, so there is only so much you can do. With an inline IDS, far more control is in your hands.

There are two prime worries with this type of configuration—false positives have even more disastrous consequences than with your average IPS, and performance can be a significant concern. Since all of your network traffic is going through this one box, a single point of failure is often worrisome from a redundancy and performance point of view.

Answering Common IDS Questions

Let's look at some of the major questions that people often have when considering an IDS for their network. It's important to understand the function of an IDS within your overall security design, the differences between an IDS and your other security devices, and what an IDS can and cannot do for you in terms of enhancing the security of your network.

Why Are Intrusion Detection Systems Important?

IDSs provide an integral audit component of a robust security design and policy. They let you know when you're being scanned and when you're being attacked. They provide more information than you could get just by checking your server and firewall logs. You can see the attacks that fail and the attacks that succeed, and get real-time notification of attempted attacks. You can watch your own network traffic and become aware of misconfigurations as well as malicious attacks earlier than you may have noticed without an IDS. They are not the be-all, end-all solution to every security woe, but they are a valuable tool in the hands of a skilled security administrator.

Why Doesn't My Firewall Serve as an IDS?

While some integrated appliances out there claim to be both a firewall and an IDS, and we are probably going to see more of those in the future, a firewall's function is to filter packets, not to alert on potentially malicious traffic. Firewalls are primarily designed to deny or allow traffic to access the network, not to alert administrators of malevolent activity. Many firewalls are only network-level packet filters, allowing or denying traffic based purely on the source and destination IP address and port. This doesn't begin to touch the complexity of the traffic analysis that an IDS handles. The simple analogy is that you don't trust the locks on your doors to also act as cameras, so why should your locks on your network (the firewalls) be expected to be cameras (the IDS)?

Why Are Attackers Interested in Me?

Put simply, because you're there. While attackers certainly do look for high-value targets (targets that have something they specifically want), any system connected to the Internet these days is a potential target. While many attackers will go for juicy-looking targets and other low-hanging fruit, not being the most tempting target out there doesn't mean you are safe. You don't want to be just a little bit more secure than the next guy... in today's digital environment, you want to be actually safe. Many managers make the mistake of thinking that the attacker wants the company's data. In most cases, the attacker wants to steal bandwidth, not secrets.

Automated Scanning/ Attacking Doesn't Care Who You Are

Many attackers scan (or even attack without scanning) entire class B subnets at a time. For those of you who don't do exponential math in your head, that's 65,536 machines at a time. Many script kiddies aren't looking for any particular machine; they just want as many compromised "zombie" machines as possible. Therefore, they will launch their automated scans, and attempt to exploit all machines that they see as vulnerable, regardless of who they are. `You@example.com` is treated just the same as `you@whitehouse.gov` or `you@google.com`. And that's more consideration than you'll get from many of the automated worms and viruses, which will happily scan random subnets and all the machines on them without any cognizance whatsoever of what machines are on those networks and whether they should be doing that after all.

So why do these attackers want so many random machines that may or may not be valuable to them? They want something you have, whether that's bandwidth, clock cycles of your CPU, or data.

Desirable Resources Make You a Target

The more you have, the more others will want it. If even Archibald Enduser is a target, larger machines and corporate networks are that much more so. But what are these miscreants hoping to do with your computer?

Bandwidth

Well-connected computers are valued in the underground for several purposes. One of the most popular is to launch distributed denial-of-service attacks (DDoS), using your bandwidth to send attack traffic to people whom they don't like. Of course, this will make your legitimate use of your computer and its network a lot slower, but they don't really care about that. Bandwidth can also be used for for-profit spambots, hijacking your computer to churn out ads for generic Viagra and plastic surgery, or for hosting high-volume warez servers of pirated software, movies, porn, and music.

Disk Space

Disk space is usually a concern for attackers planning on setting up warez servers to share out pirated software, movies, porn, and music. The more disk space you have, the more attractive your server will be to use for such purposes.

Valuable Information

If your machine has any type of sensitive information on it, it is possible that the attackers are after that. Whether it's a targeted attack to attempt to steal your secret corporate plans to build Isengard 2.0, or some attacker who got lucky, corporate espionage or information selling is not unfeasible. Look at the scandal involving partisan information theft in the U.S. Congress in 2004, for just one example.

OINK!

Because there is a profit to be made from stealing information, these attackers are frequently the best funded and most highly skilled of the threats you or any company you work for are likely to face. Case in point: Six months prior to Slammer, there was another worm that exploited a weakness in Microsoft's SQL server. The worm, known as SQL Snake, took advantage of the fact that many SQL server installations had a default SA (admin) password that was blank. The person who released the worm is said to have stolen hundreds of databases, and was offering them for sale.

Political or Emotional Motivations

Some attackers are motivated by political gain, or some sort of a feeling of revenge upon someone they don't like. The DDoS attacks generated by the MyDoom worm variants in early 2004 are an example of this, targeting sco.com and Microsoft.com, and reportedly passing over domains like google.com and Berkeley.edu. Internet Relay Chat (IRC) servers are well known in the security community for drawing fire—when Internet flamewars break out, DDoS attacks are often the result. There's a well-known ongoing series of cyber hostilities between Indian and Pakistani hackers, for example, with viruses flying back and forth and defacements proclaiming political causes and the superiority or inferiority of one nationality over the other. Since the September 11 terrorist attacks on the United States, there have been reported acts of technical jihad, with American hackers attacking sites they perceive as affiliated with al Qaeda, and vice versa.

Where Does an IDS Fit with the Rest of My Security Plan?

Alongside a good security policy, incident response plan, firewall architecture, virus checkers, and all the other features of a modern security plan for enterprise networks, an IDS can play a vital role in securing your enterprise. Your IDS can be an early warning of network trouble, often picking up malicious activity before any of your other layers of defense. Your IDS can provide necessary logs and proof of activity, should you ever need to go to court regarding a network intrusion. Your IDS can alert your system administrators and security staff to problems in time for them to take effective action, and it can be a useful tool in enforcing enterprise IT policy and flagging violations. Last but certainly not least, it can provide a warning that your other security measures may have failed in time to fix them. Many companies and organizations put a NIDS sensor on each side of their firewall and then tune the sensor on the protected side to send high-priority alerts if any traffic is seen that should not have gotten through the firewall.

Where Should I Be Looking for Intrusions?

A good security policy addresses multiple layers of security, protecting your enterprise assets in many ways. This philosophy is called “defense in depth,” and is central to mounting an effective defense against the multiple threats facing a modern enterprise. If attackers can’t get past your firewall, they may call the help desk and try to bluff them into giving away account credentials. If they can’t get in to your headquarters by walking on in, they may send your vice president an e-mail with a backdoor disguised as a holiday card. The creative ways in which attackers can approach your network are limited only by their imaginations. Unfortunately, this means that the most correct answer to this question is, “you should be looking everywhere.” However, when talking strictly about IDS placement, you should be watching every point where your network connects to another network (Internet connections, DMZs, modem banks, VPN gateways, and so forth), and any server that is important enough that you would be upset if it were compromised. If you would like to know more about some of the alternate ways that attackers use to get into companies, Kevin Mitnick’s recent book *The Art of Deception* describes some of the various nontraditional ways that security can be subverted.

Operating System Security—Backdoors and Trojans

This is the classic sort of thing that most people think of when they consider network security—Trojans, backdoors, compromises of individual boxes through weaknesses of software or configurations. In addition to good system administration practices like keeping up to date on your patching and turning off services that you don't need by default, you should consider a regular scan or vulnerability assessment of your own network. This will help you detect unknown listening services or unapproved configurations. You should have standard, documented, hardened configuration templates so that when a new machine is attached to your network, it's not going to be the gateway through which a thousand preventable compromises pour. IDSs can help greatly in watching for this type of traffic.

OINK!

There has been an interesting development from a couple of vendors (well... two so far) who are now offering software that supposedly can identify vulnerabilities on systems just by passively watching their network traffic. If it works, this would allow you to have your IDS sensor actually perform some amount of vulnerability monitoring and analysis. One of the biggest complaints many companies have with vulnerability scanning is the risk of having it crash a server or the added load on the network. This approach has the advantage of not ever touching the servers and not adding any load to the network at all. At publication time, the two vendors we know of who offer this are Tenable Security and Sourcefire.

Physical Security

Good security practices look at more than just your network connectivity. Physical attacks and approaches are alive and well. Can someone walk in to your enterprise, pick up a laptop with valuable data on it, and stroll out the door undetected? Don't laugh! This happens more often than you might imagine. It happened recently to an airline; two men dressed as technicians went in to an office and walked out with two of the company's mainframe computers. We can only speculate as to what they wanted or have done with the information they

got, since they haven't been caught. It is highly doubtful that they were just doing it for the thrill. If so, you need to give some thought to your physical security model as well as your network security. Are your servers located in a separate space with some type of access control for your staff? Any network security consultant will tell you that physical access to a device is extremely dangerous. In most cases, all you have to do is reboot the machine and set the BIOS to boot from a CD-ROM. There are security toolkits small enough to fit on a credit card-sized CD-ROM that contain all the forensics tools you'd need to discover almost any type of information about the servers' hard drives and data, and plenty that will change things at will. These toolkits are operating-system agnostic; a bootable Linux CD can reset your Administrator password for a Windows machine, for example. Even more dangerous, bootable USB drives are becoming common now, which counters the remove-all-disk-and-CDROM-drives defense.

Tools & Traps...

Bootable CD Toolkits

- **FIRE** A portable CD-ROM based Linux distribution with 196 security and forensics tools at the time of writing (version 0.4). FIRE is designed to provide an environment to do vulnerability assessment, data forensics, virus scanning, and incident response from a bootable CD-ROM. Tremendously useful to the security administrator, FIRE is also extremely useful to people of variable morality in physical vulnerability assessment scenarios. Anything you can do with this tool, an attacker can also do. Available online at <http://fire.dmzs.com/>.
- **Knoppix** A full-featured Linux environment including graphical user interface (GUI), OpenOffice, the Gimp, Abiword, and Mozilla. Less obviously useful to the attacker or the security administrator than FIRE, but offers the capability to look at office documents on the local machine right there from your own operating system, edit, and leave without having had to log in or access the system through legitimate means. Available online at www.knoppix.net.

Continued

- **Linux-BBC** Well known in the Linux community, the Linux Bootable Business Card (BBC) is a Linux distribution on CD-ROM cut to the form factor of a mini-business card. Small enough to slip into anyone's wallet unnoticed, the Linux-BBC supports large IDE disks, BitTorrent, and The Coroner's Toolkit, a software forensics package. Available online at www.linux-bbc.org.
- **Offline NT Password & Registry Editor, Bootdisk/CD** Need to change the Administrator password (or any other password) on a Windows system? Don't have a login currently? Go to <http://home.eunet.no/~pnordah/ntpasswd/bootdisk.html> and download this toolkit. In less than 10 minutes, you can change the password, boot back to Windows, and log in with your new password.

Keeping your servers away from miscreants and attackers isn't the limit of physical security, though. Guarding against someone running off with a laptop containing sensitive data, ensuring that if someone sets fire to your main data center that you have an offsite backup of all your important information, and training your staff to be aware of social engineering attempts and what to do in case of an attempted security breach are all important facets of physical security.

Application Security and Data Integrity

Are you sure that your data has not been tampered with? How do you know that the source code in your central CVS repository is the same as the source code that was there last night? How can you prove that the figures in your banking database are true and accurate rather than jimmied? Provable authentication of the integrity of your data is crucial to the modern enterprise, and there are highly motivated attackers out there just waiting to get their hands on your resources. From the attempted backdooring of the Linux source code tree in November 2003 to the wireless hack of an Israeli post office's network, leading to the alleged theft of 80,000 credit card numbers, we can see that attackers have every reason to want to take advantage of vulnerable applications. If you don't have some way of verifying that your data is unmodified or that your transactions are secure, you will be in very bad shape indeed in the event of a successful intrusion, or even a potentially successful one. Saying "I don't know" when asked about data integrity is rarely good enough for the customers.

Correlation of All These Sources

It is worth mentioning that correlating your security information from multiple sources is much more likely to help you reconstruct what happened when analyzing intrusion attempts. Data from your firewalls and routers can back up the alerts seen by your IDS. Overlapping sources can cover for each other in case of the failure of one system, and when you can correlate alerts from multiple sources, you can have a much higher confidence that you aren't dealing with a false positive. Logs of keycard swipes can help you determine who (or at the least, whose access credentials) was in a given area at the time in question, network access credentials can help you determine who logged in, and security cameras can help you verify whether the person at the keyboard was the person whose password you have on file.

What Will an IDS Do for Me?

An IDS can be a valuable addition to your security toolkit. It can give you unprecedented insight into what's really going on in your network, and alert you to new trouble or attacks before you otherwise would have seen them. It can help you monitor and enforce your company's security policies, gain deeper insight into trends in your system and network usage, and plan better for future budgeting and purchases through seeing where your blind spots and problems are. It can notify your administrators of a likely system compromise, or even of a failed attempt. And it never gets tired, never needs a coffee break, and doesn't demand a raise every time you yell at it.

Continuously Watch Packets on Your Network and Understand Them

We have yet to meet the system administrator or security engineer who is capable of this for more than five minutes, and that's on a slow network connection and generally reading hex, not binary. An IDS is perfectly capable of tirelessly matching packet after packet to its known signatures, and comparing their payloads without needing to translate into a human-readable form. Its algorithms are normally at least several orders of magnitude faster than a human attempting to perform the same job, and generally less prone to mistakes.

Read Hundreds of Megs of Logs Daily and Look for Specific Issues

An IDS can significantly speed up the amount of log files that you can parse on a daily basis. When you are responsible for the security of a large environment, the volume of log files that you'll find yourself accumulating is truly astounding (think terabytes for a large group of systems and an active high-speed network). Going over them all by hand becomes increasingly impossible the bigger your network grows. A log-parsing IDS provides a sane and sensible way to look for particular issues and signatures in your log files, giving you a better idea of what's going on with all your various devices.

Create Tremendous Amounts of Data No Matter How Well You Tune It

Even the most precisely tuned IDS is going to have voluminous output. Although it seems almost a contradiction to say so, anomalous network and system events are happening all the time. Users are becoming root. Commands are being sent over Web interfaces. Administrator passwords are being changed, packets with bad combinations of TCP flags are being sent, applications are abusing protocols in ways that only the most twisted and tortured of minds could come up with, and automated worms and viruses continue in their blind quest for self-propagation. Each of these events can trigger an IDS alert. And when you have a few thousand of them a day, well, managing your alerts becomes a major challenge.

Very often, IDS administrators are faced with the daily prospect of having to sort through a few thousand (or a few hundred thousand) alerts, many of which are known issues, but not tuned out because someone eventually intends to get around to correcting them. Some are just difficult to tune out by their very nature—many operating systems and applications send packets that just should not be! However, you can't spend your time tuning out every individual system on the Internet that might be running one of those operating systems, and you don't want to junk the signature entirely for fear of missing the actual stealthy portscans that might be network reconnaissance. When you decide to set up an IDS, be prepared for some situations akin to this to occur. No matter how well you tune, you will get data—and lots of it. Some of it will be false positives. Writing good rules and correlating your data can decrease the false positives and even the number of true positives that need to be looked at individually, but you still end up with lots of data.

Create So Much Data that If You Don't Tune It, You Might as Well Not Have It

One of our special frustrations as security geeks is encountering situations where a company has invested a fortune in the latest cutting-edge IDSs, sparing no expense, and then has hired one person with no security background whatsoever to monitor and administer them all. The poor administrator has no idea how to tune an IDS, and still less idea of how to deal with the barrage of alerts she's being hammered with. The pointy-haired boss's inevitable conclusion to this scenario is that all IDSs are worthless. After all, they paid for the best, didn't they?

Tuning the false positives out of your IDS is crucial. Having knowledgeable administrators involved in the design and placement of the sensors and then in the tuning of the ruleset is essential. If you don't know your network well enough to winnow out the known issues and the definite false positives, you'll be awash in a sea of portscans and informational alerts, with no easy way of wading through all that data to find the relatively few blatant attacks and/or subtle system compromises. Every IDS out of the box will generate massive amounts of false positives, and an unknowledgeable security geek might as well not have one.

Find Subtle Trends in Large Amounts of Data that Might Not Otherwise Be Noticed

One of the benefits of having such a massive base of data is the ability to look at trends in the alerts or packet flows. Are you getting more scans for an unusual port today than you were yesterday? Has it been steadily on the rise recently? Perhaps a new tool or exploit out there targets that port. Have you been seeing more failed logins to various servers on your network? Perhaps someone is walking around and trying to guess passwords. The ability to see the big picture in the reams of data may be enhanced by an IDS, particularly an IDS with correlation capabilities.

Supplement Your Other Protection Mechanisms

An IDS can act as confirmation or backup for your other network security systems. This goes back to the principle of defense in depth. If you are seeing exploit traffic aimed at your Web proxy and you're not sure if your proxy sanitizes the traffic before passing it on to your end user, check your IDS. See if it's alerting on the traffic both before and after the proxy. If you know that someone with Administrator access used Remote Desktop to connect to the Exchange

server right before it broke yesterday, check your IDS logs to see if you have a record of who accessed that server, from where, and (if you have both HIDSs and NIDSs) what sort of traffic he sent. The absence of an IDS alert should not be used as proof positive that everything is okay. As we said earlier, IDSs will not catch every attack. Even if they have a signature for it, a sufficiently high volume of traffic will cause the IDS to drop packets. However, the presence of an alert can be used as a backup and support to other network security systems and logs.

Act as a Force Multiplier Competent System/Network Administrator

Using an IDS, good security geeks will be able to go through far more logs and far more network data than they could without one. While an IDS will not replace additional skilled help, it can make each competent geek more effective than he would have been without the additional tools. When investigating an intrusion attempt, it is greatly helpful to be able to say, “What other alerts did this source IP or user generate? What other alerts were associated with this destination IP?” Being able to quickly put your fingers on other relevant data can help administrators understand the kind and scope of their issue far more quickly than if they had to do all the log parsing and searching by hand.

OINK!

What Is a Force Multiplier? A force multiplier is something that increases the amount of result you get back for the force exerted. Look at any book on mechanical engineering (*The Way Things Work* is a good one) for examples.

Let You Know When It Looks Like You Are Under Attack

With the myriad alerting capabilities of most IDSs out there, there are a plethora of ways to notify your on-call or on-duty system administrators when it appears that an attack is ongoing. This time saved can be an invaluable asset to an incident response team. It can make the difference between pulling one compromised system off the network before it has a chance to branch out and launch

attacks at others, or dealing with a massive enterprise-wide security breach that will take endless hours of labor to address.

What Won't an IDS Do for Me?

An IDS is not the be-all and end-all solution to all your security woes. It will not replace your system administrator, make that guy on IRC who doesn't like you go away, or answer that e-mail that you've been avoiding. It will not secure the physical perimeter of your site, magically detect every possible malicious bit flipped on your network, or tell you when one of your employees is thinking about selling you out to the competition. To get the most out of an IDS, it is important to understand its capabilities and limitations, and to design your security policy accordingly.

Replace the Need for Someone Who Is Knowledgeable about Security

Even the best IDS is only as good as its programming. It will do what you tell it to do faithfully, it will alert as you tell it to alert and, if an IPS, will respond as you tell it to respond. However, it can't tell you what to do in a new and unprecedented situation. It can't write its own signatures for new attacks, and it can't deal with an intelligent, flexible, adaptive attacker who takes an approach outside of its specifications. It cannot determine what your security policy should be. It cannot make informed recommendations for your network based on the latest industry developments. In short, it cannot replace a skilled security geek.

Catch Every Attack that Occurs

New attacks are being developed all the time. Even as we write this, even as you read this, attackers are out there trying to figure out new ways to break into systems. Sometimes these are new ways to exploit old vulnerabilities, but other times they are totally new approaches. Your IDS is not configured to handle all possible attacks, simply because some of them haven't been invented yet. You can only protect against the type of attacks of which you are aware. And even some of the attacks that are known are not guarded against by all IDSs. Your IDS will help you see the attacks and potential attacks that are out there, but it won't catch everything.

Damage & Defense...

fragroute and the Newsham/Ptacek Paper

In 1998, Tim Newsham and Tom Ptacek wrote a paper entitled, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," describing ways to evade detection by most of the IDSs then available (www.insecure.org/stf/secnet_ids/secnet_ids.pdf). The techniques in question included testing the timeouts on IDSs, checking reassembly of fragmented packets (overwriting the same data with different content), simulating delays and packet loss in network programs, and randomization of IP parameters to evade operating system fingerprinting. Although this made many people in the intrusion detection community sit up and pay attention, it was nothing compared to the stir when Dug Song released first fragrouter and later fragroute, tools that implemented most of these attacks (www.monkey.org/~dugsong/fragroute/). The theory was now reality. Many of these attacks are addressed and now detected by Snort since Snort version 1.9, but there are still many IDSs that may miss them, and some of the attacks are simply hard to address from a network perspective. One approach currently getting a lot of attention is target-based IDSs, which combine a knowledge of your network, operating systems, and configuration with live detection of attacks. The aim of target-based IDSs is to present the administrator with alerts with a tighter focus, drastically cutting the number of false positives and centering analysis on the most likely real alerts. You can read more about target-based IDSs in *Information Security Magazine* at http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss306_art540,00.html—target-based IDS reviews were featured in their cover story in January 2004.

Prevent Attacks from Occurring

No IDS out there is going to magically make attackers stop attacking you. Your defenses may prevent these attacks from succeeding, but the attackers will keep trying to break down your digital walls. No matter how good your IDS is, it will not change human nature or the inclination of malicious attackers to try to own your network.

When you are choosing and installing a NIDS, it is instructive to consider what you will not see as well as what you will see. If traffic is encrypted, you will still be able to see the IP headers and transport layer protocol headers, but you will not be able to decode the contents of the packet without breaking that encryption. You can watch how much traffic is sent, and from whom to whom, and how often, but you won't be able to see what they're saying. Depending on the type of NIDS you have deployed, this may or may not put a cramp in your style. Signature-based IDSs that depend on traffic being sent in cleartext may not alert if the traffic is encrypted. Protocol analysis may still work for encrypted traffic, but may break if the traffic is sent on an unexpected port. Traffic pattern analysis is likely to be your best bet when dealing with encryption.

OINK!

It should be obvious that your NIDS won't be able to see inside your network traffic if it is encrypted (unless you use special tools and change how you do encryption). What might not be quite as obvious is that even most HIDSs that look at network traffic (a.k.a Network Node IDS or NNIDS) won't be able to see inside encrypted traffic either. The reason for this is simply that almost all HIDSs watch network traffic as it is coming in to or going out of your system, somewhere around Layer 2 on the network stack (just before the traffic goes to the hardware from the OS). Currently, the majority of encryption is being done at the application layer (Layer 7) by applications such as your Web browser or SSH. This means that the traffic is still encrypted when the IDS sees it entering or leaving the system. This is unfortunately something that most vendors forget to mention when talking about the benefits of their products.

This is becoming more and more of a problem, as more and more environments begin using encryption in more and more of their network communications. Fortunately, IDS vendors are aware of this and are working on solutions. We hope they'll be good ones.

Prevent Attacks from Succeeding Automatically (in Most Cases)

With the exception of some IPSs, in most cases, by the time the IDS has seen the attack attempt cross the wire, it has either succeeded or it has not. In the case

of an e-mail with a viral payload, for example, it's possible that the IPS would trigger on the subject line and have time to send a reset-kill and end the mail transfer before the entire message, complete with virus, could be delivered. However, in many other cases, attack and success of the exploit follow hard on each other's heels, and there just simply isn't enough time for the IDS or IPS to jump in there between the last no-operation command and the execution of the shell code.

Replace Your Other Protection Mechanisms

While there are many all-in-one security products out there, don't be fooled into thinking that any one security product can do the job of a different type of security product. Just because you have an IDS doesn't mean that you can junk your firewall. The presence of a VPN does not mean that you don't need to patch your systems, either. The process of securing your network is aided by redundancy and layers of reinforced security. An IDS will not by itself be the only security device you'll ever need or want.

What Else Can Be Done with Intrusion Detection?

These are only some of the possible uses for an IDS. Many HIDSs allow you to audit and monitor use of shared resources. They provide enhanced capabilities of determining who is using shared network resources, provide benchmarking and resource utilization statistics for monitoring server functions, and can match subject lines or content of e-mail to be able to alert on and/or get rid of mails with known malware content. The possibilities are endless, and as flexible as your ruleset and IDS implementation.

Fitting Snort into Your Security Architecture

Since you're holding this book, we assume that you have or are interested in having Snort in your network. Snort is a very flexible network IDS, offering a multitude of rules already authored as well as the ability to write your own. There are several mailing lists where people trade new Snort rules that they've written in response to the latest attacks, and offer commentary on the rules and the new incidents they see on their networks. Snort is very full-featured, with

many preprocessors to parse different types of data, a bevy of keywords to allow matching of the content, port, protocol, and more, portscan detection, buffer length detection, and many other features—and since it's open source, you can add any functionality you like. There are also many other add-ons to support logging alerts in database formats, management and automated downloads of new rules, distribution of rules to sensors without clobbering the local rulesets, a Web interface for Snort sensor management, and others. Let's take a quick tour of Snort's usefulness in an enterprise network.

Viruses, Worms, and Snort

Within days if not hours of the release of a new worm, Snort signatures are being written for it. Those signatures are often incorporated into the main Snort ruleset, so that all Snort users can benefit from them. Signatures for SQL Slammer were out on the NANOG mailing list within hours of the initial detection of the worm (www.merit.edu/mail.archives/nanog/2003-01/msg00775.html). Signatures for the MyDoom.A worm were out within a day of the initial detects by antivirus labs. This type of quick responsiveness allows Snort users to update their rulesets when a new attack comes out, and begin detection and remediation of their vulnerabilities sooner. In fact, if you use some of the add-ons that are available for Snort, you can actually detect signs of worm propagation before signatures are available.

Known Exploit Tools and Snort

Snort has many signatures that are tailored to let you know when a known exploit tool is being used against your network. Some of these tools are marked by their self-advertising in the packet payloads, like the SolarWinds ICMP and SNMP scanner. Here's the Snort signature (www.snort.org/snort-db/sid.html?sid=1918):

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SolarWinds IP scan attempt"; content:"SolarWinds.Net"; itype:8; icode:0; classtype:network-scan; sid:1918; rev:3;)
```

Note the "SolarWinds.Net" content in the ICMP echo packet. In this case, that's the fingerprint of the tool. However, not all known exploit tools are quite so self-advertising. Consider this signature, for a Trin00 attacker client attempting to connect to the Trin00 master server on the default port with the default password:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS Trin00 Attacker  
to Master default startup password"; flow:established,to_server;  
content:"betaalmostdone"; reference:arachnids,197; classtype:attempted-dos;  
sid:233; rev:3;)
```

Although many of the Snort signatures are written as generically as possible to allow you to see the attack no matter which tool was used to generate it, the rule authors won't hesitate to write a rule for a particular tool as well if one should flag itself in a clear fashion.

Writing Your Own Signatures with Snort

It should now be obvious that one of the greatest strengths of Snort is the ability to write customized rules for your network and the traffic you see. The syntax is precise and flexible, allowing you to match all sorts of different network traffic. Additional information can be found online at www.snort.org/.

Using an IDS to Monitor Your Company Policy

A common use of customized Snort rules is to monitor traffic that, while not actively malicious, is restricted or frowned upon by company policy. Some enterprises write rules to alert them when their users access a Web page with content matching particular keywords, or a site with unauthorized software, or other policy violations. Snort actually comes with a set of rules for traffic that is likely to be pornography. You can even write your own Snort rules to match any type of network traffic, letting you know when someone has shut down the mail server and started up the Quake server.

Analyzing Your IDS Design and Investment

Once you have decided which type(s) of IDSs you want to deploy and where you'd like to place them in your network, it's time to give some thoughtful consideration to how you might improve your design. Are you likely to be inundated with false alerts, or miss alerts you would like to see? Could a real attack slip by in the midst of a storm of false positives?

False Positives versus False Negatives

When trying to establish an IDS policy, one expects to be inundated with false positives; at least until some IDS tuning has been done to get them down to a manageable roar. More concerning, however, is the possibility of false negatives, those attacks that the IDS misses. It is all too easy to be lulled into a false sense of security—seeing many alerts every day often gives us the impression that since we're seeing so many potential attacks, surely we must be seeing them all. However, skilled attackers can scan and code their exploits specifically to be stealthy and not detected. There are a variety of techniques available for doing this, which we will discuss.

Fooling an IDS

The Ptacek & Newsham paper previously mentioned discusses many individual techniques for fooling a NIDS, but in general, there are two main approaches. One approach is to give it so much data that it chokes on it, either missing packets or drowning the administrator in so many alerts that she never sees the real attack. The other general approach is to frame your attack in such a way that it won't match the signatures or algorithms that the IDS is using to pull out the attacks from the network background noise. The former technique is what the tools Stick and Snot depend on, as well as Nmap's decoy scan. The latter technique is what the stealth Nmap scans and tools like Dug Song's fragrouter or Rain Forest Puppy's Whiskeruse.

IDS Evasion Techniques

First, let's look at the noisy way. Stick and Snot (see the sidebar) are tools designed to generate as many alerts as possible on your IDS. They do this by generating alerts from a ruleset that is likely similar to the ruleset your IDS is using to match traffic. Some miscreants hope to slip in some attack traffic while you're distracted by all the false positives, or while your IDS is dropping packets. Others just like the idea of killing your IDS.

If the attacker used Stick or Snot to cover his tracks and then launched a TCP attack, this could be easily compensated for by only having Snort alert on established TCP sessions. However, this would be an ideal time for the attacker to launch a UDP-based attack—Remote Procedure Call (RPC), DNS, something like that.

For maximum stealth, the attacker could even spoof the source; that doesn't matter in connectionless UDP. There is some likelihood that the attack packets would get dropped if the network links were too oversaturated with the Stick/Snot output, but it is likely that the actual attack packets would not be picked up by the IDS, either because it's only listening to established TCP sessions and our attack is UDP or ICMP, or because the IDS is still listening to all connections but is mobbed with false positives.

Notes from the Underground...

Stick, Snot, and Snort

Stick, Snot, and Snort are tools billed as "IDS Killers," designed to overload your IDS to the point it becomes unusable.

- **Stick** (www.eurocompton.net/stick/projects8.html) is a C program based on an old version of the Snort ruleset, designed to spew out so many alert-triggering packets per second that it would force IDSs to come to a grinding halt. It was very effective for its time, but Snort now has measures in place to adjust to and compensate for this style of attack.
- **Snot** is another similar tool (www.stolenshoes.net/sniph/index.html) that takes a Snort ruleset as argument and generates a series of packets that will trigger that ruleset. Cross-platform and flexible, Snot allows script kiddies all over the world to annoy to their IDS administrators.

If your Snort installation is being harried by these tools or similar ones, you can limit your Snort alerts to noticing established TCP sessions only with the *snort -z est* arguments. For this to work, however, the stream4 preprocessor must be configured. Also keep in mind that this will limit you from seeing all other nonstateful TCP alerts, so you will be missing UDP, ICMP, and ARP-based alerts. However, your IDS will still be up and running.

Nmap offers a noisy scan that generates a whole bunch of fake packets as alternate “sources,” using the `-D` “decoy” option. To the target, it looks like they are being scanned by all the decoy machines at once, and your real scan is masked among the fake ones.

Now, the quiet way. These are the attackers you really need to worry about. We have already described fragroute and Dug Song’s evasive techniques as laid out in the original Newsham-Ptacek paper, but Nmap also offers options for stealth. There is the idle scan, the FTP bounce attack, timing-based attacks like a very slow scan stretched out over days, fragmentation and reassembly based attacks, TCP flag combination attacks, and even an idle scan off an unwitting zombie host. To read details about the packet construction behind all these attacks, refer to the Nmap man page at www.insecure.org/nmap/data/nmap_manpage.html.

Return on Investment—Is It Worth It?

At the end of the day, the deciding factor for many businesses is what the expected return on investment is. Is there truly going to be enough enhancement to your network security that it’s worth installing, configuring, and maintaining an IDS? Security is often referred to as an economic sinkhole for businesses; they spend money on it, but if all goes well, they rarely see returns. Instead, the returns are in costs saved rather than in products made. Because of this, many CEOs are reluctant to spend the money necessary for expensive systems or solutions, more so if they’ve already spent money on an IDS and have seen few positive results from it but many false positives.

If you are considering adding an IDS to your network, consider it as a business case. How much money does your company lose if there is an intrusion? What are the odds of that intrusion happening? How much will it cost to install and maintain an IDS? How much will the IDS offset or mitigate the risks of that intrusion? How will an IDS affect your organization legally? Earlier in the appendix, we discussed the possible implications of wiretap and privacy laws on a company’s use of an IDS. However, an IDS can also assist in compliance with corporate accounting laws such as the Sarbanes-Oxley requirements, and in establishing an audit trail in the event of a compromise. Sections 302 and 304 of the Sarbanes-Oxley requirements place the responsibility on a corporation to establish internal controls within their network. An IDS can be a demonstrable part of these controls. When combined with a third-party penetration test of your network security, this can go a long way toward validating your own data

with an external audit, complete with trail. Some locations now require companies to notify customers when their data has been compromised; the State of California is one such place. Having an IDS can allow you to detect compromise attempts more reliably. Being able to go to your CEO with strong numbers, legal backing, and business precedent will be far more impressive than “uh, I guess we need one of those, everyone else seems to have one.”

Defining IDS Terminology

Being able to understand the differences between different types of IDSs and their features is crucial when trying to design a security architecture. Let's look at some of the most common terminology in the IDS field, and make sure we understand all the options available.

Intrusion Prevention Systems (HIPS and NIPS)

An IDS that not only detects possible attack, but also responds to prevent the attack from being successful. This response can be anything from creating firewall rules to black-hole the attacker, to killing the offending process (when dealing with a Host IPS), to dropping the offending traffic (when dealing with a Network IPS).

Gateway IDS

An IDS that sits at the bottleneck between your network and the Internet (or whatever peering upstream you may be connected to). Also known as an inline IDS, all traffic must pass through this gateway to leave your local network. This may also function as an IPS if it includes the capability to make decisions about whether traffic should be allowed.

Network Node IDS

The method of intrusion detection where one establishes a baseline of “normal” network traffic, and then looks for deviations from that norm and flags them as possible attack traffic.

Protocol Analysis

The method of intrusion detection where one looks at the flow of data within the specifications of each protocol, looking for anomalies and possible malicious traffic based on the expected protocol behavior.

Target-Based IDS

A new flavor of IDSs specifically aimed at what is actually on the network. They are designed to have fewer false positives and only alert on attacks that are relevant to your network and the specific services running on your network.

Summary

IDSs can serve many purposes in a defense-in-depth architecture. In addition to identifying attacks and suspicious activity, you can use IDS data to identify security vulnerabilities and weaknesses.

IDSs can audit and enforce security policy. For example, if your security policy prohibits the use of file-sharing applications such as Kazaa, Gnutella, or messaging services such as Internet Relay Chat (IRC) or Instant Messenger, you could configure your IDS to detect and report this breach of policy.

IDSs are an invaluable source of evidence. Logs from an IDS can become an important part of computer forensics and incident-handling efforts. Detection systems are used to detect insider attacks by monitoring traffic from Trojans or malicious code and can be used as incident management tools to track an attack.

Correlation of data, whether from a HIDS or NIDS or DIDS, is probably the best way to approach intrusion detection data. While an IDS can be a valuable contributor to a security architecture, it is by no means enough in and of itself to protect a network.

A NIDS can be used to record and correlate malicious network activities. The NIDS is stealthy and can be implemented to passively monitor or to react to an intrusion. The HIDS plays a vital role in a defense-in-depth posture; it represents the last bastion of hope in an attack. If the attacker has bypassed all of the perimeter defenses, the HIDS might be the only thing preventing total compromise. The HIDS resides on the host machine and is responsible for packet inspection to and from that host only. It can monitor encrypted traffic at the host level, and is useful for correlating attacks that are detected by different network sensors. Used in this manner it can determine whether the attack was successful. The logs from a HIDS can be a vital resource in reconstructing an attack or determining the severity of an incident.

Solutions Fast Track

Introducing Intrusion Detection Systems

- ☑ An intrusion is an unauthorized access, use, or attack on your network or computers.
- ☑ IDSs work by watching network and system activity, and comparing that to known signatures or against algorithms to separate legitimate activity from suspicious activity.

- ☑ IDSs can then log the attack and respond in a number of ways. The most common response is to alert the system administrators through SNMP traps, text messages, phone calls, or pages.

Answering Common IDS Questions

- ☑ Attackers are interested in everyone connected to the Internet these days; it's not necessarily personal.
- ☑ An IDS can alert you to network traffic and system activity of which you may not have been aware. It can increase the effectiveness of a good system administrator, and provide him with additional data.
- ☑ An IDS will not replace your existing security staff, or make people stop attacking you.

Fitting Snort into Your Security Policy

- ☑ Snort is a network IDS with sophisticated pattern-matching capabilities that are used to uniquely describe attack traffic.
- ☑ Snort signatures for the latest viruses, worms, and other new vulnerabilities are usually written and released within hours or days of the new attacks' debut.
- ☑ You can write your own Snort signatures to match company policy violation, new or unique traffic, or anything else.

Analyzing IDS Design and Architecture

- ☑ IDSs can be configured to just detect and alert, or to respond as well.
- ☑ Possible responses include dropping the traffic, spoofing ICMP or TCP Reset packets, or identifying and tracing back toward the attack source.
- ☑ IDSs are not perfect or foolproof—they can be tricked or eluded. They are valuable contributors to a security policy, but not enough all by themselves to enforce it.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this appendix and to assist you with real-life implementation of these concepts. To have your questions about this appendix answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Why doesn't my firewall serve as an IDS?

A: Firewalls are designed primarily to pass, drop, or reject traffic, not to alert on suspicious traffic. IDSs are designed to let you know when suspicious activity is occurring. The two functions are different and conflict in key issues.

Q: Can IDSs gather data from anywhere besides sniffing on a network?

A: Yes, some IDSs can also gather data from log parsing, watching system calls, or monitoring a filesystem.

Q: What can an IDS do for me that my system administrator can't?

A: Parse a few hundred million packets or log entries (or more) a day in binary. Most administrators get tired after a while.

Q: What can my system administrator do for me that my IDS can't?

A: Bring creative thinking and an understanding of the significance of this network activity to the analysis.

Q: Will I have to spend time tuning my IDS?

A: Yes. If you don't want to be drowning in false positives, it really is best to tune your IDS to fit its environment.

Q: Does physical security still matter if I have the best network security in the world?

A: Absolutely. If we can walk in to your office and walk out with your server, you've still been rooted.

Q: Why should I bother writing my own signatures, when Snort has so many already?

A: You certainly don't have to, but you might want to add functionality that's not present in the extant ruleset, like rules tailored to your enterprise policy or to detect attacks targeting specific proprietary applications.

Introducing Vulnerability Assessments and Nessus

Solutions in this Appendix:

- What Is a Vulnerability Assessment?
- Automated Assessments
- Two Approaches
- Realistic Expectations

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

In the war zone that is the modern Internet, manually reviewing each networked system for security flaws is no longer feasible. Operating systems, applications, and network protocols have grown so complex over the last decade that it takes a dedicated security administrator to keep even a relatively small network shielded from attack.

Each technical advance brings wave after wave of security holes. A new protocol might result in dozens of actual implementations, each of which could contain exploitable programming errors. Logic errors, vendor-installed backdoors, and default configurations plague everything from modern operating systems to the simplest print server. Yesterday's viruses seem positively tame compared to the highly optimized Internet worms that continuously assault every system attached to the global Internet.

To combat these attacks, a network administrator needs the appropriate tools and knowledge to identify vulnerable systems and resolve their security problems before they can be exploited. One of the most powerful tools available today is the vulnerability assessment, and this appendix describes what it is, what it can provide you, and why you should be performing them as often as possible. Following this is an analysis of the different types of solutions available, the advantages of each, and the actual steps used by most tools during the assessment process. The next section describes two distinct approaches used by the current generation of assessment tools and how choosing the right tool can make a significant impact on the security of your network. Finally, the appendix closes with the issues and limitations that you can expect when using any of the available assessment tools.

What Is a Vulnerability Assessment?

To explain vulnerability assessments, we first need to define what a vulnerability is. For the purposes of this book, *vulnerability* refers to any programming error or misconfiguration that could allow an intruder to gain unauthorized access. This includes anything from a weak password on a router to an unpatched programming flaw in an exposed network service. Vulnerabilities are no longer just the realm of system crackers and security consultants; they have become the enabling factor behind most network worms, spyware applications, and e-mail viruses.

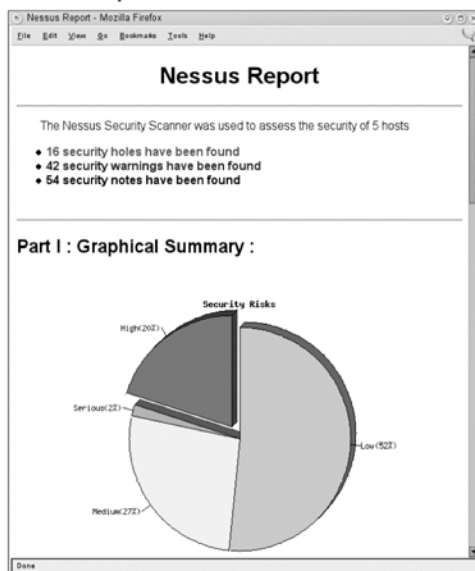
Spammers are increasingly relying on software vulnerabilities to hide their tracks; the open mail relays of the 1990s have been replaced by compromised “zombie” proxies of today, created through the mass exploitation of common

vulnerabilities. A question often asked is, “Why would someone target my system?” The answer is that most exploited systems were not targeted; they were simply one more address in a network range being scanned by an attacker. They were targets of opportunity, not choice. Spammers do not care whether a system belongs to an international bank or your grandmother Edna; as long as they can install their relay software, it makes no difference to them.

Vulnerability assessments are simply the process of locating and reporting vulnerabilities. They provide you with a way to detect and resolve security problems before someone or something can exploit them. One of the most common uses for vulnerability assessments is their capability to validate security measures. If you recently installed a new intrusion detection system (IDS), a vulnerability assessment allows you to determine how well that solution works. If the assessment completes and your IDS didn’t fire off a single alert, it might be time to have a chat with the vendor.

The actual process for vulnerability identification varies widely between solutions; however, they all focus on a single output—the report. This report provides a snapshot of all the identified vulnerabilities on the network at a given time. Components of this report usually include a list detailing each identified vulnerability, where it was found, what the potential risk is, and how it can be resolved. Figure C.1 shows a sample Nessus Security Scanner report for a network of only five systems; the number of vulnerabilities is already over 100!

Figure C.1 Sample Nessus Report



Why a Vulnerability Assessment?

Vulnerability assessments have become a critical component of many organizations' security infrastructures; the ability to perform a networkwide security snapshot supports a number of security vulnerability and administrative processes. When a new vulnerability is discovered, the network administrator can perform an assessment, discover which systems are vulnerable, and start the patch installation process. After the fixes are in place, another assessment can be run to verify that the vulnerabilities were actually resolved. This cycle of assess, patch, and re-assess has become the standard method for many organizations to manage their security issues.

Many organizations have integrated vulnerability assessments into their system rollout process. Before a new server is installed, it first must go through a vulnerability assessment and pass with flying colors. This process is especially important for organizations that use a standard build image for each system; all too often, a new server can be imaged, configured, and installed without the administrator remembering to install the latest system patches. Additionally, many vulnerabilities can only be resolved through manual configuration changes; even an automated patch installation might not be enough to secure a newly imaged system. It's much easier to find these problems at build time when configuration changes are simple and risk-free than when that system is deployed in the field. We strongly recommend performing a vulnerability assessment against any new system before deploying it.

While many security solutions complicate system administration, vulnerability assessments can actually assist an administrator. Although the primary purpose of an assessment is to detect vulnerabilities, the assessment report can also be used as an inventory of the systems on the network and the services they expose. Since enumerating hosts and services is the first part of any vulnerability assessment, regular assessments can give you a current and very useful understanding of the services offered on your network. Assessments assist in crises: when a new worm is released, assessment reports are often used to generate task lists for the system administration staff, allowing them to prevent a worm outbreak before it reaches critical mass.

Asset classification is one of the most common nonsecurity uses for vulnerability assessment tools. Knowing how many and what types of printers are in use will help resource planning. Determining how many Windows 95 systems still need to be upgraded can be as easy as looking at your latest report. The ability to glance quickly at a document and determine what network resources might be overtaxed or underutilized can be invaluable to topology planning.

Assessment tools are also capable of detecting corporate policy violations; many tools will report peer-to-peer services, shared directories full of illegally-shared copyrighted materials, and unauthorized remote access tools. If a long-time system administrator leaves the company, an assessment tool can be used to detect that a backdoor was left in the firewall. If bandwidth use suddenly spikes, a vulnerability assessment can be used to locate workstations that have installed file-sharing software.

One of the most important uses for vulnerability assessment data is event correlation; if an intrusion does occur, a recent assessment report allows the security administrator to determine how it occurred, and what other assets might have been compromised. If the intruder gained access to a network consisting of unpatched Web servers, it is safe to assume that he gained access to those systems as well.

Notes from the Underground...

Intrusion Detection Systems

The difference between vulnerability assessments and an IDS is not always immediately clear. To understand the differences between these complementary security systems, you will also need to understand how an IDS works. When people speak of IDSs, they are often referring to what is more specifically called a network intrusion detection system (NIDS). A NIDS' role is to monitor all network traffic, pick out malicious attacks from the normal data, and send out alerts when an attack is detected. This type of defense is known as a *reactive security measure* as it can only provide you with information after an attack has occurred. In contrast, a vulnerability assessment can provide you with the data about a vulnerability before it is used to compromise a system, allowing you to fix the problem and prevent the intrusion. For this reason, vulnerability assessments are considered a *proactive security measure*.

Assessment Types

The term *vulnerability assessment* is used to refer to many different types and levels of service. A host assessment normally refers to a security analysis against a single

system, from that system, often using specialized tools and an administrative user account. In contrast, a network assessment is used to test an entire network of systems at once.

Host Assessments

Host assessment tools were one of the first proactive security measures available to system administrators and are still in use today. These tools require that the assessment software be installed on each system you want to assess. This software can either be run stand-alone or be linked to a central system on the network. A host assessment looks for system-level vulnerabilities such as insecure file permissions, missing software patches, noncompliant security policies, and outright backdoors and Trojan horse installations.

The depth of the testing performed by host assessment tools makes it the preferred method of monitoring the security of critical systems. The downside of host assessments is that they require a set of specialized tools for the operating system and software packages being used, in addition to administrative access to each system that should be tested. Combined with the substantial time investment required to perform the testing and the limited scalability, host assessments are often reserved for a few critical systems.

The number of available and up-to-date host assessment solutions has been decreasing over the last few years. Tools like COPS and Tiger that were used religiously by system administrators just a few years ago have now fallen so far behind as to be nearly useless. Many of the stand-alone tools have been replaced by agent-based systems that use a centralized reporting and management system. This transition has been fueled by a demand for scalable systems that can be deployed across larger server farms with a minimum of administrative effort. At the time of this publication the only stand-alone host assessment tools used with any frequency are those targeting nontechnical home users and part-time administrators for small business systems.

Although stand-alone tools have started to decline, the number of “enterprise security management” systems that include a host assessment component is still increasing dramatically. The dual requirements of scalability and ease of deployment have resulted in host assessments becoming a component of larger management systems. A number of established software companies offer commercial products in this space, including, but not limited to, Internet Security System’s System Scanner, Computer Associates eTrust Access Control product line, and BindView’s bvControl software.

Network Assessments

Network assessments have been around almost as long as host assessments, starting with the Security Administrator Tool for Analyzing Networks (SATAN), released by Dan Farmer and Wietse Venema in 1995. SATAN provided a new perspective to administrators who were used to host assessment and hardening tools. Instead of analyzing the local system for problems, it allowed you to look for common problems on any system connected to the network. This opened the gates for a still-expanding market of both open-source and commercial network-based assessment systems.

A network vulnerability assessment locates all live systems on a network, determines what network services are in use, and then analyzes those services for potential vulnerabilities. Unlike the host assessment solutions, this process does not require any configuration changes on the systems being assessed. Network assessments can be both scalable and efficient in terms of administrative requirements and are the only feasible method of gauging the security of large, complex networks of heterogeneous systems.

Although network assessments are very effective for identifying vulnerabilities, they do suffer from certain limitations. These include: not being able to detect certain types of backdoors, complications with firewalls, and the inability to test for certain vulnerabilities due to the testing process itself being dangerous. Network assessments can disrupt normal operations, interfere with many devices (especially printers), use large amounts of bandwidth, and create fill-up disks with log files on the systems being assessed. Additionally, many vulnerabilities are exploitable by an authorized but unprivileged user account and cannot be identified through a network assessment.

Automated Assessments

The first experience that many people have with vulnerability assessments is using a security consulting firm to provide a network audit. This type of audit is normally comprised of both manual and automated components; the auditors will use automated tools for much of the initial legwork and follow it up with manual system inspection. While this process can provide thorough results, it is often much more expensive than simply using an automated assessment tool to perform the process in-house.

The need for automated assessment tools has resulted in a number of advanced solutions being developed. These solutions range from simple graphical user inter-

face (GUI) software products to stand-alone appliances that are capable of being linked into massive distributed assessment architectures. Due to the overwhelming number of vulnerability tests needed to build even a simple tool, the commercial market is easily divided between a few well-funded independent products and literally hundreds of solutions built on the open-source Nessus Security Scanner. These automated assessment tools can be further broken into two types of products: those that are actually obtained, through either purchase or download, and those that are provided through a subscription service.

Stand-Alone vs. Subscription

The stand-alone category of products includes most open-source projects and about half of the serious commercial contenders. Some examples include the Nessus Security Scanner, eEye's Retina, Tenable Security's Lightning Proxy, and Microsoft's Security Baseline Scanner. These products are either provided as a software package that is installed on a workstation, or a hardware appliance that you simply plug in and access over the network.

The subscription service solutions take a slightly different approach; instead of requiring the user to perform the actual installation and deployment, the vendor handles the basic configuration and simply provides a Web interface to the client. This is primarily used to offer assessments for Internet-facing assets (external assessments), but can also be combined with an appliance to provide assessments for an organization's internal network. Examples of products that are provided as a subscription service include Qualys' QualysGuard, BeyondSecurity's Automated Scan, and Digital Defense's Frontline product.

The advantages of using a stand-alone product are obvious: all of your data stays in-house, and you decide exactly when, where, and how the product is used. One disadvantage, however, is that these products require the user to perform an update before every use to avoid an out-of-date vulnerability check set, potentially missing recent vulnerabilities. The advantages of a subscription service model are twofold: the updates are handled for you, and since the external assessment originates from the vendor's network, you are provided with a real-world view of how your network looks from the Internet.

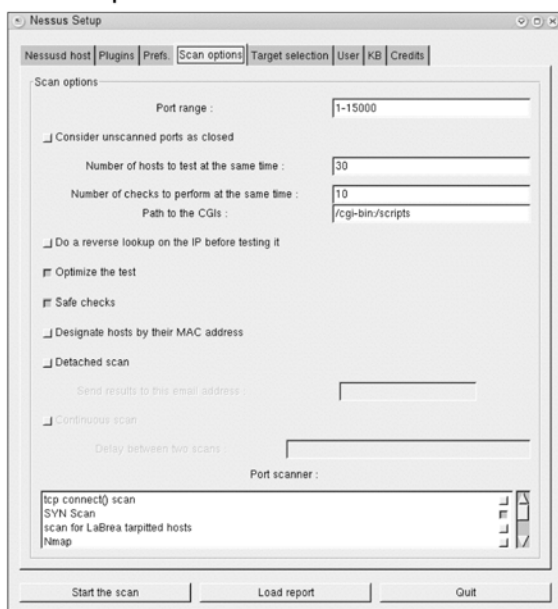
The disadvantages to a subscription solution are the lack of control you have over the configuration of the device, and the potential storage of vulnerability data on the vendor's systems. Some hybrid subscription service solutions have emerged that resolve both of these issues through leased appliances in conjunction with user-provided storage media for the assessment data. One product that

implements this approach is nCircles' IP360 system, which uses multiple dedicated appliances that store all sensitive data on a removable flash storage device.

The Assessment Process

Regardless of what automated assessment solution is used, it will more than likely follow the same general process. Each assessment begins with the user specifying what address or address ranges should be tested. This is often implemented as either a drop-down list of predefined ranges or a simple text widget where the network address and mask can be entered. Once the addresses are specified, the interface will often present the user with a set of configuration options for the assessment; this could include the port ranges to scan, the bandwidth settings to use, or any product-specific features. After all of this information is entered, the actual assessment phase starts. Figure C.2 shows the assessment configuration screen for the Nessus Security Scanner.

Figure C.2 Nessus Scan Options



Detecting Live Systems

The first stage of a network vulnerability assessment determines which Internet Protocol (IP) addresses specified in the target range actually map to online and accessible systems. For each address specified by the user, one or more probes are

sent to elicit a response. If a response is received, the system will place that address in a list of valid hosts. In the case of heavily firewalled networks, most products have an option to force scan all addresses, regardless of whether a response is received during this stage.

These types of probes sent during this stage differ wildly between assessment tools; although almost all of them use Internet Control Message Protocol (ICMP) “ping” requests, the techniques beyond this are rarely similar between two products. The Nessus Security Scanner has the capability to use a series of TCP connection requests to a set of common ports to identify systems that might be blocking ICMP messages. This allows the scanner to identify systems behind firewalls or those specifically configured to ignore ICMP traffic. After a connection request is sent, any response received from that system will cause it to be added to the list of tested hosts. Many commercial tools include the capability to probe specific User Datagram Protocol (UDP) services in addition to the standard ICMP and TCP tests. This technique is useful for detecting systems that only allow specific UDP application requests through, as is commonly the case with external DNS and RADIUS servers.

Identifying Live Systems

After the initial host detection phase is complete, many products will use a variety of fingerprinting techniques to determine what type of system was found at each address in the live system list. These fingerprinting techniques range from Simple Network Management Protocol (SNMP) queries to complex TCP/IP stack-based operating system identification.

This stage can be crucial in preventing the assessment from interfering with the normal operation of the network; quite a few print servers, older UNIX systems, and network-enabled applications will crash when a vulnerability assessment is performed on them. Indeed, the biggest problem that most administrators encounter with automated assessment tools is that they can disrupt network operations. Often, the administrator will have to spend time rebooting devices, retrieving garbage printouts from network-attached print servers, and debugging user problems with network applications. This identification stage can often be used to detect and avoid problematic systems before the following stages can cause problems.

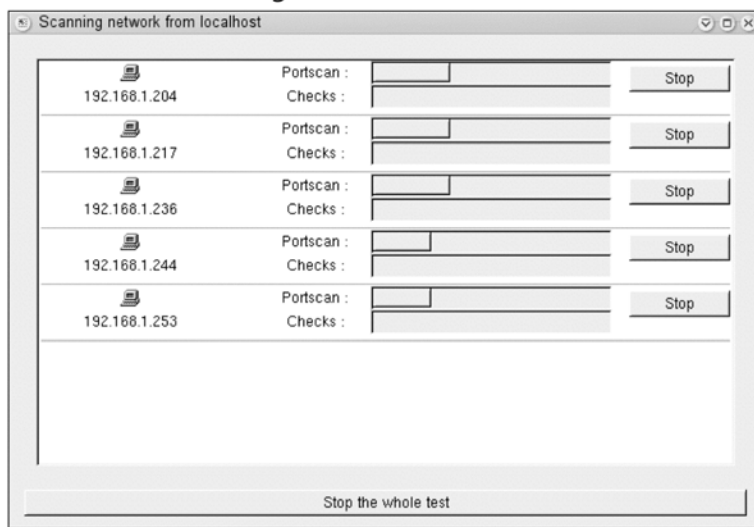
Enumerating Services

Once the host detection and identification steps are complete, the next stage is normally a port scan. A port scan is the process of determining what TCP and

UDP services are open on a given system. TCP port scans are conducted by sending connection requests to a configured list of port numbers on each system. If the system responds with a message indicating that the port is open, the port number is logged and stored for later use. UDP port scanning can often provide inconsistent results, since the nature of the protocol makes obtaining consistent results difficult on most networks.

There are 65,536 available TCP ports; however, most assessment tools will only perform a port scan against a limited set of these. Limiting the scan to a subset of the available ports reduces the amount of time it takes to perform the assessment and substantially decreases the bandwidth required by the assessment (in terms of packets per second, not the total number of bytes). The downside of not scanning all available ports is that services that are bound to nonstandard, high port numbers are often completely ignored by the assessment. The Nessus Security Scanner provides an option that allows the user to define how these ports are treated. The default is to consider all nonscanned TCP ports open, which can take quite a bit of time during the assessment, especially in cases where heavy packet filters or firewalls are in place. Figure C.3 shows the Nessus Security Scanner performing the service enumeration phase of the assessment.

Figure C.3 Nessus Enumerating Services



Identifying Services

After the port scan phase, many assessment tools will try to perform service identification on each open port. This process starts with sending some common application requests and analyzing the responses against a set of signatures. When a signature matches a known application, this information is stored for the later use and the next service is tested. Although not all assessment tools perform this stage, the ones that do can provide much more accurate results, simply by knowing which vulnerabilities to check for on what ports.

The Nessus Security Scanner includes a robust service identification engine, capable of detecting more than 90 different application protocols. This engine uses a set of application probes to elicit responses from each service. After each probe is sent, the result is matched against a list of known application signatures. When a matching signature is found, the port number and protocol are stored for future use and the engine continues with the next service. If the Secure Sockets Layer (SSL) transport protocol is detected, the engine will automatically negotiate SSL on the service before sending the application probes. This combination of transport-level and service-level identification allows the system to accurately detect vulnerabilities even when the affected service is on a nonstandard port.

The HyperText Transfer Protocol (HTTP) is a great example of a service that is often found on a port other than the default. Although almost all standard Web servers will use TCP port 80, literally thousands of applications install an HTTP service on a port other than 80. Web configuration interfaces for many Web application servers, hardware devices, and security tools will use nonstandard ports. E-mail protocols such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), and Internet Message Access Protocol (IMAP) are often configured with the SSL transport protocol and installed on nonstandard ports as well. A common misconfiguration is to block spam relaying on the primary SMTP service, but trust all messages accepted through the SSL-wrapped SMTP service on a different port. Additionally, this phase prevents an application running on a port normally reserved for another protocol from being ignored completely by the scan or resulting in false positives.

Identifying Applications

Once the service detection phase is complete, the next step is to determine the actual application in use for each detected service. The goal of this stage is to identify the vendor, type, and version of every service detected in the previous stage. This information is critical, as the vulnerability tests for one application can

actually cause another application to crash. An example of this is if a Web server is vulnerable to a long pathname overflow. If any other vulnerability tests send a request longer than what is expected by this system, the application will crash. To accurately detect this vulnerability on the Web server instead of crashing it, the system must first identify that specific application and then prevent any of the problematic vulnerability tests from running against it.

One of the most common problems with most assessment tools is that of the *false positive* where the tool reports a vulnerability that does not actually exist on the tested systems. False positives can produce a huge amount of verification work for the assessment engineer. When application identification information is either missing or incomplete, test results will often include false positives. When the developers of these assessment tools write the vulnerability tests, they often assume that the system they are interacting with is always going to be the product in which the vulnerability was discovered. Different applications that offer the same service will often respond to a probe in such a way that the vulnerability test logic registers a vulnerability. For this reason, application identification has become one of the most critical components of modern assessment tools.

Identifying Vulnerabilities

After every online host has been identified, each open port has been mapped to a known service, and the known services have been mapped to specific applications, the system is finally ready to begin testing for vulnerabilities. This process often starts with basic information-gathering techniques, followed by active configuration probes, and finally a set of custom attacks that can identify whether a particular vulnerability exists on the tested system.

The vulnerability identification process can vary from simple banner matching and version tests, to complete exploitation of the tested flaw. When version detection and banner matching are used to identify a vulnerability, false positives often result due to application vendors providing updated software that still displays the banner of the vulnerable version. For this reason, version numbers are often consulted only when there is no other way to safely verify whether the vulnerability exists.

Many common vulnerabilities can only be identified by attempting to exploit the flaw. This often means using the vulnerability to execute a command, display a system file, or otherwise verify that the system is indeed vulnerable to an attack by a remote intruder. Many buffer overflow and input manipulation vulnerabilities can

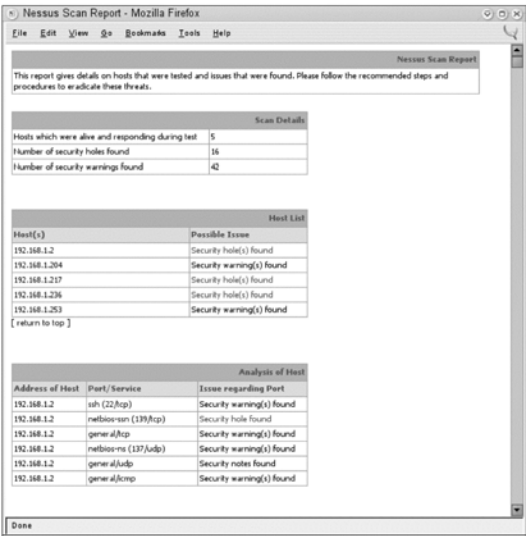
be detected by triggering just enough of the flaw to indicate that the system has not been patched, but not enough to actually take down the service. The assessment tool has to walk a fine line between reliable vulnerability identification and destructive side effects.

Vulnerability tests that use banner checks will encounter problems when the tested service has been patched, either by the vendor or system administrator, but the version number displayed to the network has not been updated, or at least when it has not been updated in the way the vulnerability test expects. This is a relatively common practice with open-source UNIX-based platforms and certain Linux distributions.

Reporting Vulnerabilities

After the analysis is finished, the final stage of the assessment process is reporting. Each product has a unique perspective on how reports should be generated, what they should include, and in what formats to provide them. Regardless of the product, the assessment report will list the systems discovered during the assessment and any vulnerabilities that were identified on them. Many products offer different levels of reporting depending on the audience; it is useful to provide a high-level summary to management giving a system administrator a report that tells him or her what systems need to be fixed and how to do so. One of the popular features in many assessment tools is the capability to show trend reports of how a given network fared over time. Figure C.4 shows the Nessus Security Scanner's HTML report summary.

Figure C.4 Nessus Report Summary



Two Approaches

When performing an automated vulnerability assessment, the actual perspective of the test can have a huge impact on the depth and quality of the results. Essentially, there are two different approaches to vulnerability testing: administrative and outsider. Each has distinct advantages and disadvantages, such that many of the better assessment tools have migrated to a hybrid model that combines the best features of both approaches. Understanding these different approaches can provide insight into why two different assessment tools can provide such completely different results when used to test the same network.

Administrative Approach

The administrative approach performs the assessment from the perspective of a normal, authenticated system administrator. The assessment tool might require that it be launched by an authenticated administrative user or provided with a user account and password. These credentials can be used to detect missing patches, insecure configuration settings, and potentially vulnerable client-side software (such as e-mail clients and Web browsers).

This is a powerful approach for networks that consist of mostly Windows-based systems that all authenticate against the same domain. It combines much of the deep analysis of a host assessment with the network assessment's scalability advantages. Since almost all of the vulnerability tests are performed using either remote registry or remote file system access, there is little chance that an assessment tool using this method can adversely affect the tested systems. This allows assessments to be conducted during the day, while the systems are actively being used, without fear of disrupting a business activity.

The administrative approach is especially useful when trying to detect and resolve client-side vulnerabilities on a network of workstations. Many worms, Trojans, and viruses propagate by exploiting vulnerabilities in e-mail clients and Web browser software. An assessment tool using this approach can access the registry of each system and determine whether the latest patches have been installed, whether the proper security settings have been applied, and often whether the system has already been successfully attacked. Client-side security is one of the most overlooked entry points on most corporate networks; there have been numerous cases of a network with a well-secured perimeter being overtaken by a network simply because a user visited the wrong Web site with an outdated Web browser.

Unfortunately, these products often have some severe limitations as well. Since the testing process uses the standard Windows administrative channels—namely, the NetBIOS services and an administrative user account—anything preventing this channel from being accessed will result in inaccurate scan results. Any system on the network that is configured with a different authentication source (running in stand-alone mode, on a different domain, or authenticating to a Novell server) will not be correctly assessed. Additionally, these products may have issues similar to the issues of host-based assessment tools, network devices, UNIX-based servers, and IP-enabled phone systems may also be completely missed or return incomplete results.

Network and host-based firewalls can also interfere with the assessment. This interference is a common occurrence when performing assessments against a system hosted on a different network segment, such as a demilitarized zone (DMZ) or external segment behind a dedicated firewall. Additionally, network devices, UNIX-based servers, and IP-enabled phone systems might also be either completely missed or have only minimal results returned. An example of this is a certain Windows-based commercial assessment tool that will report missing Internet Information Server (IIS) patches even when the Web server has not been enabled or configured.

This type of testing is very helpful to verify a networkwide patch deployment, but should not be relied upon as the only method of security testing. Microsoft's Security Baseline Scanner is the best example of an assessment tool that uses this approach alone. Many of the commercial assessment tool offerings were originally based on this approach and have only recently started to integrate different techniques into their vulnerability tests. The differences between administrative and hybrid solutions is discussed at length in the section *The Hybrid Approach*.

The Outsider Approach

The outsider approach takes the perspective of the unauthenticated malicious intruder who is trying to break into the network. The assessment process is able to make decisions about the security of a system only through a combination of application fingerprinting, version identification, and actual exploitation attempts. Assessment tools built on this approach are often capable of detecting vulnerabilities across a much wider range of operating systems and devices than their administrative approach counterparts can.

When conducting a large-scale assessment against a network consisting of many different operating systems and network devices, the outsider approach is

the only technique that has a chance of returning accurate, consistent results about each discovered system. If a system is behind a firewall, only the exposed services will be tested, providing you with the same information that an intruder would see in a real-life attack. The reports provided by tools that use this hybrid approach are geared to prevent common attacks; this is in contrast to those tools using the administrative approach that often focus on missing patches and insecure configuration settings. In essence, the outsider approach presents a much more targeted list of problems for remediation, allowing the administrator to focus on the issues that would be the first choices for a potential intruder.

Although this approach is the only plausible method of conducting a vulnerability assessment on a heterogeneous network, it also suffers from a significant set of drawbacks. Many vulnerabilities simply cannot be tested without crashing the application, device, or operating system. The result is that any assessment tools that test for these types of vulnerabilities either provide an option for “intrusive” testing, or always trigger a warning when a potentially vulnerable service is discovered. Since the outsider approach can only detect what is visible from the point in the network where the assessment was launched, it might not report a vulnerable service bound to a different interface on the same system. This is an issue with reporting more than anything else, as someone reviewing the assessment report might not consider the network perspective when creating a list of remediation tasks for that system.

The Hybrid Approach

Over the last few years, more and more tools have switched to a hybrid approach for network assessments. They use administrative credentials when possible, but fall back to remote fingerprinting techniques if an account is either not available or not accepted on the tested system. The quality of these hybrid solutions varies greatly; the products were originally designed with only the administrative approach in mind have a difficult time when administrative credentials are not available, whereas the products based on the outsider approach often contain glitches when using an administrative account for tests. It seems that the latter has better chances at overcoming its hurdles without requiring a re-write. Overall, though, these products provide results that are often superior to those using a single approach. The Nessus Security Scanner and eEye’s Retina product are examples of tools that use this approach.

One of the greatest advantages of tools using the outsider approach is that they are often able to determine whether a given vulnerability exists, regardless of

whether a patch was applied. As many Windows network administrators know, installing an operating system patch does not actually guarantee that the vulnerability has been removed. A recent vulnerability in the Microsoft Windows Network Messenger service allowed a remote attacker to execute arbitrary code on a vulnerable system. Public exploits for the vulnerability started circulating, and companies were frantically trying to install the patch on all their internal workstations. Something that was overlooked was that for the patch to take effect, the system had to be rebooted after it was applied. Many sites used automated patch installation tools to update all their vulnerable systems, but completely forgot about the reboot requirement.

The result was that when an assessment was run using a tool that took the administrative approach, it reported the systems as patched. However, when an assessment was run using the Nessus Security Scanner, it reported these systems as vulnerable. The tool using the administrative approach simply checked the registry of each system to determine whether the patch had been applied, whereas the Nessus scan actually probed the vulnerability to determine if it was still vulnerable. Without this second assessment, the organization would have left hundreds of workstations exposed, even though the patches had been applied. The registry analysis used by many tools that take the administrative approach can miss vulnerabilities for a number of other reasons as well. The most common occurrence is when a hotfix has been applied to resolve a vulnerability, and then an older service pack is reapplied over the entire system. The changes installed by the hotfix were overwritten, but the registry entry stating that the patch was applied still exists. This problem primarily affects Windows operating systems; however, a number of commercial UNIX vendors have had similar issues with tracking installed patches and determining which ones still need to be applied.

Recently, many of the administrative and hybrid tools have developed new techniques for verifying that an installed patch actually exists. Shavlik Technology's HFNetChk Pro will actually check the last reboot time and compare it to the hotfix install date. The Nessus Security Scanner actually accesses the affected executables across the network and verifies the embedded version numbers.

The drawbacks to the hybrid approach are normally not apparent until the results of a few large scans are observed; because the administrative approach is used opportunistically, vulnerabilities that are reported on a system that accepts the provided user account might not be reported on a similar system that uses a different authentication realm. If the administrator does not realize that the other system might be vulnerable as well, it could lead to a false sense of security. These missed vulnerabilities can be difficult to track down and can fall under the radar

of the administrator. Because there is a higher chance of these systems not being patched, the hybrid approach can actually result in more damage during an intrusion or worm outbreak. Although the administrative approach suffers from the same issue, tools using the administrative approach take it for granted that systems outside of the authentication realm will not be tested.

Realistic Expectations

When the first commercial vulnerability assessment tools started becoming popular, they were advertised as being able to magically identify every security hole on your network. A few years ago, this might have been close to the truth. The number of publicly documented vulnerabilities was still quite small, and tracking vulnerability information was an obscure hobby. These days, the scenario is much different, whereas there were a few hundred well-documented vulnerabilities before, there are literally thousands of them now, and they don't even begin to scratch the surface when it comes to the number of flaws that can be used to penetrate a corporate network.

In addition to the avalanche of vulnerabilities, the number and type of devices found on an average corporate network has exploded. Some of these devices will crash, misbehave, or slow to a crawl during a network vulnerability assessment. A vulnerability test designed for one system might cause another application or device to stop functioning altogether, annoying the users of those systems and potentially interrupting the work flow. Assessment tools have a tough job; they have to identify as many vulnerabilities as possible on systems that must be analyzed and categorized on the fly, without reporting false positives, and at the same time avoid crashing devices and applications that simply weren't designed with security in mind. Some tools fare better than others; however, all current assessment tools exhibit this problem in one form or another.

When someone first starts to use a vulnerability assessment system, he or she often notices that the results between subsequent scans can differ significantly. This issue is encountered more frequently on larger networks that are connected through slower links. There are quite a few different reasons for this, but the core issue is that unlike most software processes, remote vulnerability testing is more of an art form than a science. Many assessment tools define a hard timeout for establishing connections to a service or receiving the result of a query. If an extra second or two of latency occurs on the network, the test could miss a valid response. These types of timing issues are common among assessment tools; however, many other factors can play into the consistency of scan results.

Many network devices provide a Telnet console that allows an administrator to reconfigure the system remotely. These devices will often set a hard limit on the number of concurrent network connections allowed to this service. When a vulnerability assessment is launched, it might perform multiple tests on a given port at the same time; this can cause one check to receive a valid response, while another gets an error message indicating that all available connections are being used. If that second check was responsible for testing for a default password on this particular device, it might completely miss the vulnerability. If the same scan was run later, but the default password test ran before one of the others, it would accurately detect the vulnerability at the expense of the other tests. This type of timing problem is much more common on network devices and older UNIX systems than on most modern workstations and servers, but can ultimately lead to inconsistent assessment results.

Tools & Traps...

Assessing Print Servers

Almost all vulnerability assessment tools have one thing in common; they are capable of eating a print server alive. The problem stems from the fact that many print servers offer a variety of network services that can be used to spool documents directly to the attached printer. The most problematic of these services is the Direct Print Protocol, which is a TCP service. This can cause problems with automated assessment tools, as the service identification phase can often cause reams of paper to be printed out, covered in what appears to be garbage. Another common issue relates to the custom FTP service that many print servers run. This service will allow authentications using any username and password combination and simply prints out any files that are uploaded. If the assessment tool is looking for insecure FTP configurations, it might end up printing out a test file when running against a print server. To compound matters, quite a few print servers have such shoddy TCP/IP implementations that a simple port scan can take them offline, and a full power cycle is required to return them to service.

Dynamic systems are the bane of the vulnerability assessment tools. If an assessment is in full swing and a user decides to reboot his workstation, the assessment tool will start receiving connection timeouts for the vulnerability tests. Once the

system comes back online, any subsequent tests will run normally; however, all tests launched during the period of downtime will result in missing vulnerability results for that system. This type of problem is incredibly difficult to detect when wading through a massive assessment report, and at this time only a handful of commercial systems offer the capability to detect and rescan systems that restart during the assessment process.

Despite the extraordinary amount of refinement and testing that most assessment tools have undergone, false positives continue to annoy network administrators and security consultants alike. As we discussed earlier in the appendix, a false positive is simply a vulnerability that is reported, but does not actually exist on the tested system. These annoyances can build to quite a bit of verification work—before you throw out Nessus or any vulnerability assessment application for the false positive load, take the time to tune it. Nonstandard Web servers, backported software packages, and permissive match strings inside vulnerability test scripts are the top causes for false positives.

The Web server software that provides a configuration console for many network devices is notorious for causing false positives; instead of returning a standard “404” error response for nonexistent files, these systems will often return a success message for any file that is requested from the system. In response, almost all of the popular assessment tools have developed some form of Web server fingerprinting that allows their system to work around these strange Web servers. These solutions range from incredibly robust, such as the one found in the recent versions of the Nessus Security Scanner, to almost not worth the bother, as in certain commercial products.

The Limitations of Automation

Vulnerability assessment tools are still no replacement for a manual security audit by a team of trained security experts. Although many assessment tools will do their best to find common vulnerabilities in all exposed services, relatively simple vulnerabilities are often missed. Custom web applications, written under tight deadlines and for small user bases, often perform inadequate security checks on user input, but automated assessment systems may not find these flaws. Although the chances of an automated assessment tool being able to find a vulnerability in this software are slim, a security analyst experienced with Web application testing could easily pinpoint a number of security issues in a short period of time. Just because an automated assessment does not find any vulnerabilities does not mean that none exist.

Summary

As the number of discovered vulnerabilities increases every day, networks are becoming increasingly difficult to keep secure. Vulnerability assessments have become the preferred method of managing security flaws for many organizations. The ability to quickly identify misconfigured and unpatched systems, combined with the ease of use and accuracy of many assessment tools, has changed the way many administrators manage their systems. Network vulnerability assessments provide the wide view of security weaknesses on a given network, supplemented by host assessment solutions that provide granular hardening steps for critical systems.

The traditional process of system hardening and patch application has been left in the dust; as the sheer quantity of vulnerabilities is more than most administrator teams can keep track of, especially for diverse networks. Automated assessment solutions have come to the rescue, with both stand-alone and subscription-based options. The average administrator no longer needs to become a security savant simply to keep his or her systems secure. The same repeatable process allows administrators to track, resolve, and verify vulnerabilities.

Although almost all assessment tools advertise their capability to detect and report all critical vulnerabilities, the way these systems are designed and the techniques they use for vulnerability tests vary widely. Not all assessment solutions are created equal; tools using the administrative approach are almost useful when it comes to identifying vulnerabilities in network devices and across large networks. At the same time, tools using the outsider approach are restricted by the technical limitations of the vulnerabilities themselves, often ignoring vulnerabilities that they simply are unable to test. Fortunately, many of the more popular solutions have solidified around a hybrid approach for vulnerability testing, allowing for unprecedented levels of accuracy and depth.

Vulnerability assessments are not a security panacea; although they excel at detecting vulnerabilities in widely deployed products, even relatively simple flaws can be missed. The current market of assessment tools can often cause problems with network devices, slow internetwork links, and custom applications. No matter what tool you use, false positives will always be a significant problem; although many solutions have made huge steps in reducing these, backported patches and vague version identifiers will guarantee that these never entirely disappear. The depth and flexibility of a manual security assessment will always be better than any automated solution; there is no replacement for a skilled analyst manually reviewing your systems, network architecture, and in-house applications.

Solutions Fast Track

What Is a Vulnerability Assessment?

- ☑ A vulnerability is any flaw that an attacker can use to gain access to a system or network.
- ☑ Vulnerability assessments provide a snapshot of the security posture of your network.
- ☑ Host assessments provide detailed information about the weaknesses on a system.
- ☑ Network assessments pinpoint flaws that a remote attacker can use to gain access.

Automated Assessments

- ☑ Manual assessments are no longer feasible for entire networks due to the sheer number of vulnerabilities that exist.
- ☑ Stand-alone and subscription assessment models each have distinct advantages.
- ☑ Automated assessments tend to follow the same process regardless of the tool.
- ☑ The assessment process is essentially staged information gathering.

Two Approaches

- ☑ Two assessment tools can provide very different results depending on their approach.
- ☑ The administrative approach is often safest, but might not be reliable.
- ☑ The outsider approach provides the same information an attacker would have.
- ☑ Robust assessment tools use a hybrid approach for maximum vulnerability coverage.

Realistic Expectations

- ☑ Assessments can cause a myriad of side effects on an average corporate network.
- ☑ Consecutive between assessments is often less than ideal.
- ☑ False positives will always be an issue, but recent tools are making progress.
- ☑ Manual security audits still provide better results than any assessment tool can.
- ☑ Penetration testing can provide a deeper, if not wider, view of your network, from the perspective of an attacker.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this appendix and to assist you with real-life implementation of these concepts. To have your questions about this appendix answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- Q:** I am planning to use a vulnerability assessment tool at my organization. Is there any reason to assess the internal networks as well as the external?
- A:** While systems exposed to the Internet should always be incorporated into a vulnerability assessment plan, internal assessments can actually reduce the risk to the organization even more. When a new worm appears that exploits one or more known vulnerabilities, the first step an organization should take is to secure all external and internal systems. An internal assessment can be used to verify that internal assets are not at risk to an automated attack. Internal networks are vulnerable to infection through users who are compromised through their e-mail clients and Web browsers; a worm infection on an internal network segment can result in the inability for the business to function. Additionally, unethical consultants, disgruntled employees, and visitors using the network can leverage insecure systems to gain access to sensitive information.

Q: What is the difference between a vulnerability assessment and a penetration test?

A: One of the biggest problems with the security industry is consistent naming of services. A strong contributing fact is that many near dishonest security firms are selling “penetration tests” that are nothing more than a vulnerability assessment using automated tools. A vulnerability assessment is the process of identifying vulnerabilities on a network, whereas a penetration test is focused on actually gaining unauthorized access to the tested systems and using that access to the network or data, as directed by the client. A penetration test is a great way to determine how well your security measures respond to a real-life attack and what an attacker could accomplish or compromise, but may not result in a detailed analysis of every system on your network.

Q: Can a vulnerability assessment find users with weak passwords?

A: Although manual vulnerability assessments can include password auditing, automated vulnerability assessment tools are rarely able to detect common or weak passwords. The reason behind this is not that the tool is not technically able to perform the check, but that the process of testing each user could result in an account lockout. This is primarily the case with Windows domains; however, it can also apply to many commercial UNIX systems. While some automated assessment tools will test for accounts with a default or blank password, they would still not be able to detect an account with a simple one-character password. Finally, automatic tools might slow the application or network being tested. This is a part of the security assessment process that needs to be very carefully coordinated with administrators to achieve maximum success while causing a minimum of negative effects for users.

Q: My organization uses an intrusion prevention system (IPS). What complications will this cause with a vulnerability assessment?

A: The goal of an IPS is to block hostile traffic before it reaches a potentially vulnerable system. Many automated assessment solutions depend on being able to send a specially crafted attack probe and to determine whether the system is vulnerable by analyzing the response. If the IPS blocks the initial probe, the vulnerability assessment will not be able to accurately detect that vulnerability. The solution to this is either to configure the IPS to specifically ignore traffic originating from the vulnerability assessment tool, or only run

the tool from the protected side of the IPS. Most assessment tools are not designed to bypass these systems; however, an advanced intruder could easily detect the IPS and find a way to exploit a vulnerability while avoiding the IPS's block. Evading intrusion detection and prevention could easily be a book of its own; however, sufficient it to say that what the IPS is looking for might not be what the intruder uses to successfully exploit the vulnerability.

Index

A

access control
 See also passwords, permissions
 with EXT2/3, 113
 and file systems, 106
ACL (access control lists), 118, 143
Acme Widgets
 asset listings for (table), 297
 authentication solutions for,
 82–84
 company profile, 3–4
 current infrastructure diagram, 7
 functional requirements
 document (table), 9–10
 Internet Mail Service
 configuration, 184–185
 migrate-directory configuration
 file, 91–92
 test plan, 14–16
Active Directory
 DNS server support, 45
 migrating, 88, 93–97
 understanding, 58–59
Adaptec 2940UW card, 131
administration
 and network analysis, 369–378
 remote. *See* remote
 administration
administrators, system. *See* system
 administrators
Adobe Acrobat Reader plug-in,
 349–350
Advanced Intrusion Detection
 Environment (AIDE), 424
Aethera e-mail application,
 333–334
AfterSTEP window manager, 326
aliases, Apache, 271–272
AMANDA (Advanced Maryland
 Automatic Network Disk
 Archiver)
 and backup, 125
 backup suite, functions, 129–132
 generally, 146
 installing, 132–138
amanda.conf, 135
amreport script (AMANDA), 130
Analyzer sniffer, 374–375
analyzing
 DHCP traffic, 28
 IDS design, investment, 446–450
 vulnerabilities. *See* vulnerability
 assessment
Andreson, Marc, 251
anonymous access, Web services,
 257
anti-spam, anti-virus services
 applications, 219–221
 designing services, 221–225

 integrating services with
 messaging systems,
 212–225, 232–233
AntiSniff, 394
Apache
 aliases, scriptaliases, 256, 271–272
 changing default Web page, 270
 graphical administration tools,
 284–285
 installing, 264
 migrating static sites from IIS,
 286
 migrating Web services from IIS,
 250–251
 modules, controlling, 282–284
 overriding configuration
 directives, 275–276
 security and permissions,
 272–274
 virtual hosting support, setting
 up, 276–282
Apache Web servers
 configuring, 265–270
 described, using, 262–263
 summary, 287
applications
 anti-spam, anti-virus, 219–221
 common file formats (table), 299
 Horde project (table), 239
 keeping secure, 436
 Linux equivalents for Windows,
 311–312
 migrating data, profiles, 305–306
 Office suites, 350–356
 Outport (Outlook migration),
 243–244
 running Windows on Linux,
 356–359
 searching for specific, 316
appointment-making, standard, 342
architecture
 Internet e-mail, designing,
 205–208
 LAMP, 247
archives, star, tar, 122
archiving backup media, 127–129
arp command, 379
ARP redirects, 389, 393
assessing
 company's current infrastructure,
 2–8
 current desktop environment,
 294–300, 314
 file formats, 299
 print servers, 474
 types of workers, 294–295
 vulnerabilities. *See* vulnerability
 assessment
assets lists
 creating desktop, 295–299
 described, 292

 and order of tasks, 301
 and vulnerability assessment, 458
attachments, blocking for virus
 protection, 218–219
attacks
 See also specific attack
 defeating switches, 389–391
 detecting. *See* IDSs
 determining what constitutes,
 441
 DoS, 431
 organizations to help mitigate,
 410
 preventing, 442–444
 rogue DHCP servers, 30
 sniffer. *See* sniffers, sniffing
 Trojan horse, 216
 and VPNs, 402
 vulnerability assessment. *See*
 vulnerability assessment
authconfig command, 78, 88
authentication
 designing Linux-based services,
 81–88
 LDAP. *See* LDAP authentication
 managing with configuration
 files, 76–78
 migrating Windows logon files,
 97
 Network Information Services
 (NIS), 76
 services. *See* authentication
 services
 user, and PAM, 79–81
 Windows 98/NT/2000/XP
 logon process, 73–74
 Windows-based methods, 257
authentication services
 designing cross-platform, 82–84
 introduction to, 72
 migrating to Linux
 infrastructure, 17
Automachron, 36–37
automated vulnerability
 assessments, 461–468

B

back doors
 and operating system security,
 434
 and sniffers, 371
Back Orifice, 371
backing up
 AMANDA process, 130–132
 desktop systems, 306–307
 EA/ACL file system metadata,
 121
 file backup, restore, replication
 options, 123–129
Backup Domain Controller (BDC),
 57

- Ballystyx Engineering
 - authentication solutions for, 82–84
 - company profile, 4–5
 - current infrastructure diagram, 7
 - designing directory services, 61–64
 - migrate-directory configuration file, 95–96
 - migrate-smbauth configuration file, 94–95
- BASH shell, 312
- basic authentication, 257, 274
- Bayesian analysis and fighting spam, 215
- BDC (Backup Domain Controller (BDC)), 57
- Behlendorf, Brian, 263
- Berkeley (BSD)-based printing system, 152, 153
- Berners-Lee, Tim, 250, 251
- Berstein, Daniel, 198
- BIND
 - Active Directory support, 45
 - and DHCP, configuring for DDNS, 33–34
- bind operations, 55–56, 66
- Blackbox desktop, 322–323
- Blackbox window manager, 319, 326
- blacklisting spammers, 216
- bookmarks, migrating, 347
- broadcast domains, 386
- broadcast protocols, 378
- browsers. *See* Web browsers
- BSD-based print systems, 152, 154
- buffer overflow attacks, 411
- C**
- cable taps, 385, 391
- calendaring services
 - Aether's, 333–334
 - Evolution, 331
 - introduction to, 236
 - Linux-based, 238–242
 - migrating to Linux, 242–244
 - Outlook, 237
- Carnivore network analyzer, 375–378
- Carrier Sense Multiple Access/Collision Detection. *See* CSMA/CD
- CD/DVD media
 - as backup media, 126
 - and desktop backups, 306
- cd00r back door sniffer, 372
- CERT/CC security advisory, 409
- Certificate Authority (CA), 194
- certificate requests, 259, 279–280
- certificates, server, 98
- changing
 - default Web page, 270
 - passwords in Apache, 274
 - passwords regularly, 82
- channel bonding, 140
- CHAP (Challenge Handshake Authentication Protocol), 344
- chapter summaries
 - authentication services, 100
 - desktop environment, applications, 359
 - desktop migration roadmap, 314
 - directory services, 67
 - groupware, calendaring services, 245
 - IDSs (Intrusion Detection Systems), 452
 - messaging services, 230
 - network analysis, 398
 - network service migration roadmap, 18
 - TCP/IP networking services, 46
 - Web services, 287
- check disk (CHKDSK), 111
- Check Promiscuous Mode (CPM), 395
- CIFS (Common Internet File Services), 106
- ClamAV virus scanner, 220–221
- Clark, Jim, 251
- clients
 - e-mail, 190
 - Linux. *See* Linux clients
 - Linux NTP, 35–36
 - Windows time service, 36–37
- clock skew, 34–35
- Code Weaver's CrossOver Office, 302, 348, 362
- CodeRed worm, 408
- collision domains, 386–387
- .com Internet domain names, 69
- commands
 - See also specific commands*
 - UNIX printing (tables)
- Common Internet File Services (CIFS), 106
- Common UNIX Print System. *See* CUPS
- compact disks (CDs) as backup media, 126
- company Directory Information Trees (DITs), 53–54
- company profiles
 - Acme Widgets, 3–4
 - Ballystyx Engineering, 4–5
- computers, training, 304
- configuration files
 - 2wlmnt-directory-auth, 91–92
 - Acme Widgets migrate-smbauth, 90–91
 - amanda.conf, 135
 - /etc/ldap.conf (LDAP), 78
 - /etc/pam.conf (PAM), 79
 - nsswitch.conf (LDAP), 77
 - smb.conf. *See* smb.conf
- configuring
 - Apache Web servers, 265–270
 - BIND, 33–34, 42
 - Courier-IMAP, 193
 - DHCP and DNS services with Webwin, 49
 - DHCP servers, 28–31
 - IIS virtual servers, 260–262
 - IP-based virtual hosting, 277–278
 - ISC DHCPD server, 41
 - Linux print services, 153–161
 - printer options, 161
 - Samba, 84–88
 - tape drives, 131
 - and testing OpenLDAP servers, 64–66
 - connecting to directory servers, 55–56
 - connectors, Exchange 5.5, 184
 - containers
 - described, using, 266
 - VirtualHost, 267
 - control flags (PAM), 81
 - Copy In and Out (cpio) and backup media, 128
 - costs
 - Return On Investment. *See* Return On Investment (ROI)
 - Windows to Linux desktop migration, 316
 - Courier-analog log analyzer, 204
 - Courier-IMAP, 192, 193, 227–228
 - Courier-MLM mailing list manager, 204
 - Courier MTA messaging, 197, 201–204
 - Courier suite, messaging software, 200–201
 - cpio command, 128
 - creating
 - desktop assets list, 295–299
 - functional requirements document, 8–10
 - new users, Apache servers, 274
 - test plan, 14–16
 - CrossOver Office, 302, 348, 362
 - CSMA/CD (Carrier Sense Multiple Access/Collision Detection), 384–385
 - CUPS (Common UNIX Print System)
 - configuring Linux printing using, 155–161
 - migrating Windows print services to Samba, 175
 - printing style (table), 166
 - remote administration of, 178
 - and UNIX printing, 152
 - cupsd.conf, 178
 - Custom Recipients, 58
 - customizing
 - groupware solutions, 240
 - slapd, 69
 - Cyrus-IMAP, 192

D

daemons, OpenLDAP, 60
 data integrity, protecting, 436
 DDNS (Dynamic DNS),
 configuring BIND and
 DHCP for, 33–34
 demilitarized zones (DMZs), 470
 denial-of-service attacks, 431
 deploying Linux desktops, 312–313
 designing
 anti-spam, anti-virus services,
 221–225
 Linux-based authentication
 services, 81–88
 Linux-based directory services,
 61–66, 69
 Linux-based file services,
 139–140, 147
 Linux-based messaging services,
 205–212
 Linux desktop, 314
 Linux infrastructure, 11–13
 migration to Linux desktop, 292
 OpenLDAP infrastructure, 62–63
 post-migration infrastructure,
 11–13
 desktop environments
 alternative, 327–329
 assessing current, 295–299
 common, 318–323
 Device Manager, gathering desktop
 information with, 298
 DHCP clients and DHCP leases,
 24–27
 DHCP (Dynamic Host Control
 Protocol), 22–23, 40–41
 DHCP Manager, 38
 DHCP servers
 configuring, 28–31
 obtaining, releasing DHCP lease,
 24–28
 dhcpd.conf, 28, 33
 DHCPDP, open source DHCP
 server, 28
 Dia diagramming program, 6
 diagrams
 Linux e-mail server, 211–212
 post-migration infrastructure
 design, 12, 13
 digest authentication, 257, 274
 directories
 home, Windows and Linux, 309
 home, Samba network, 86–87
 structure, 53–54
 virtual, 256
 Directory Information Trees (DITs)
 described, using, 53–54
 designing for Linux-based
 directory service, 61–62
 directory schemas, 55
 directory servers, connecting to,
 55–56

directory services
 Exchange 5.5, 58
 introduction to, 52
 Linux-based, designing, 61–66
 summary, 67–68
 disk drives, partitions on, 106
 displaying
 MAC addresses, 379
 Samba printer properties, 171
 Distributed Intrusion Detection
 System (DIDS), 416–418
 Distribution Lists and mailing lists,
 58
 DITs (Directory Information
 Trees), 53–54, 61–62
 DLT tapes as backup media, 126
 DNS (Domain Name Service)
 and network concepts, 22
 understanding, 31–34
 DNS Zone Transfers, 43–44
 documenting
 functional requirements
 specification, 8–10, 300
 instructions for new desktop
 users, 293
 Linux desktop questions, 313
 server and application
 configurations, 13
 Domain Name Service. *See* DNS
 (Domain Name Service)
 domains
 collision, 386–387
 Samba NT4-style, and Active
 Directory, 94
 Windows NT, 58, 59
 download sites
 AMANDA, 129
 Courier-IMAP, 193
 Fedora Core Linux, 307
 FILEACL, 143
 Gentoo file manager, 341
 open source applications, 316
 packet traces, 397
 sniffer detectors, 394
 downloading printer drivers,
 170–174
 dpkg command, 264
 drivers, printer, 150, 159, 170–174
 DSL (Digital Subscriber Line) and
 Internet e-mail, 205
 Dsniff network sniffer, 374, 391
 dumptcpl (AMANDA), 132, 136
 Dynamic DNS (DDNS),
 configuring BIND and
 DHCP for, 33–34
 Dynamic Host Control Protocol
 (DHCP), 22–23, 40–41

E

e-mail
 choosing client, 336–337, 360
 determining storage method, 210
 encryption support, 194
 and Exchange 5.5 directory
 services, 58
 ham, and spam, 213
 and messaging systems, 182
 migrating, 337–342
 migrating to Linux, 228–229
 protecting, 396
 sending and receiving, 187–189
 testing functionality, 14
 undeliverable, 199
 viruses. *See* viruses
 EFS (Encrypting File System), 111
 eGroupware suite, 238–239
 emulator software, 361–362
 emulators, and Windows
 applications, 356
 enabling
 encryption, 97–99
 printers, 153–154
 encrypted passwords, 76, 82
 Encrypting File System (EFS), 111
 encryption
 e-mail client, server support, 194
 enabling, 97–99
 Encrypting File System (EFS),
 111
 and network performance, 401
 password, 76, 82
 protecting against sniffers, 395
 Enhanced Meta File (EMF), 150,
 155
 Enlightenment window manager,
 325
 enterprise messaging services, 182
 /etc/fstab (file system table, Linux),
 112
 /etc/ldap.conf, 78, 103
 /etc/openldap/ldap.conf, 103
 /etc/pam.conf, 79
 /etc/passwd/shadow
 authentication, 76
 Ethernet Network Analyzer, 373,
 367
 Ethernet
 and CSMA/CD, 384–385
 sniffing and, 378–380
 EtherPeek network analyzer, 374
 Ettercap sniffer, 374
 Evolution e-mail client, 330, 339
 Exchange 5.5
 directory services, 58
 Evolution's Connector for, 332
 groupware, calendaring features,
 245–246
 messaging features, component
 services, 182–185
 migrating, 88–93
 protocol support (table), 185
 Exchange servers, Outlook features
 with (table), 236–237
 Exim MTA messaging software,
 200
 exporting mailbox objects, 90

Ext2/3 file system, 112–115

F

failover clustering, 186
 FAT (File Allocation Table) file
 systems vs. NTFS 4.0, 110
 FAT16, 109
 FAT32, 108
 FBI's Carnivore network analyzer,
 375–378
 Fedora Project, 263
 feedback forms for pilot group
 testing, 16
 fetchmail, 195
 File Allocation Table (FAT) file
 systems, 106–109
 file backup options, 123–129, 146
 file-based name resolution, 31–32
 file formats, cataloging, 299
 file permissions, Apache, 273–274
 file services
 Linux, 112–117
 Windows, 106–112
 file systems
 and intrusion detection, 423–424
 Linux, 145, 310
 Minix, 113
 NTFS. *See* NTFS file system
 Windows, 145, 310
 FILEACL, reclusively dumping
 ACLs for local and remote
 shares, 143
 files
 and attachments likely to spread
 viruses, 218–219
 which to back up, 124–125
 filters for fighting spam, 215
 fingerprinting, spam, 214–215
 FIRE security tools, 435
 Firefox browser, 344–345
 firewalls and IDSs, 430, 454
 firewire drives, 126
 flooding switch, 389
 force multiplier, 440
 FQDM (fully qualified domain
 name), 136
 frames, and network traffic, 381
 Freshmeat.net, 301, 316
 fsck error checking programs,
 112–113, 117
 FTP (File Transfer Protocol) and
 OSI model, 380
 'Full Control' permissions, 109
 fully qualified domain name
 (FQDM), 136
 functional requirements document
 example, 304
 functional requirements
 specifications
 creating functional requirements
 document, 300
 desktop environment, 294
 FVWM window manager, 326

G

Galeon browser, 345–346
 gateway IDSs, 450
 gateways, anti-spam, anti-virus,
 222–224
 General Public License (GPL)
 ClamAV virus scanner, 220
 Gnome desktop, 320
 Generic Security Services
 Application Programming
 Interface (DSSAPI), 56
 Gentoo file manager, 341
 getfacl command, 119
 Ghostscript, 155
 GIF (Graphics Interchange Format)
 printer output, 151
 GIMP image editor, 298
 GLIBC v.2.3, 112
 Global Positioning Services (GPS)
 and NTP functionality, 35
 Gnome applications, running in
 KDE, Blackbox
 environments, 362
 Gnome desktop
 described, using, 284, 309,
 319–321
 Galeon browser, 345–346
 GNU C Library (glibc), 77
 GNU tar, 128
 GPOs (Group Policy Objects), 74
 Graham, Robert, 378
 Graphics Device Interface (GDI),
 150
 Graphics Interchange Format
 (GIF), 151
 'Grimm's Ping,' 407
 Group Policy Objects (GPOs) and
 Windows 2000 Active
 Directory, 74
 groupware
 eGroupware, 238–239
 introduction to, 236
 Linux-based, 238–242
 migrating to Linux, 242–244
 open source applications, 247
 GSSAPI (Generic Security Services
 Application Programming
 Interface) and LDAP
 authentication, 75

H

Hack Proofing Your Network
 (Graham), 378
 hackers and rogue DHCP servers,
 30
 ham, and spam, 213
 Hancom Office suite, 355–356
 hardware
 asset lists, 296–299
 of network analyzers, 368
 and physical security, 435
 Help, getting, 312
 HFNetChk Pro, 472

HIDS (host-based IDSs), 414
 home directories
 Windows and Linux, 309
 Samba network, 86–87
 HoneyNet Project, 397
 Horde Application Framework,
 using, 239–210
 host-based IDSs (HIDSs), 415–416
 host names and IP addresses, 31–32
 hosts, assessing vulnerabilities, 459
 .htaccess files (Apache), 275–276
 HTML (Hypertext Markup
 Language) and Courier-
 analog, 204
 htpasswd command, 274
 HTTP (Hypertext Transfer
 Protocol)
 introduction to, 250–251
 and OSI model, 380
 summary, 287
 HTTPD configuration tool
 (Apache), 284
 httpd.conf, 265–268
 hubs
 and collision domain, 401
 described, 386

I

Ical calendaring standard, 342
 ICMP redirects, router
 advertisements, 390
 IDE (Integrated Development
 Environments), 295
 IDEALX smbldap tools, 86, 88
 IDSs (Intrusion Detection Systems)
 analyzing design, investment in,
 446–450
 application security, data
 integrity, 436
 distributed (DIDS), 416–418
 function described, 424–429
 function of, 411–415
 host-based (HIDSs), 415–416
 importance of, 430–432
 introduction to, 404–411
 matching with your needs,
 437–441
 network (NIDS), 412–414
 operating system, physical
 security, 434–436
 packet sniffing, 422
 policy-based, 421
 and security plans, 432–433
 terminology, 450–451
 vs. vulnerability assessment, 459,
 479–480
 ifconfig command, 384, 391
 Ifstatus anti-sniffing program, 395
 IIS (Internet Information Server)
 default Web directory, 289
 default Web site, 255
 described, 287
 introduction, default index page,
 252–254

- migrating static sites to Apache, 286, 287
 - image editing with GIMP, 298
 - IMAP (Internet Mail Access Protocol)
 - e-mail storage, 210
 - enabling services, 226
 - protocol, and messaging, 192
 - index page, default (IIS), 252–253
 - information resources
 - Active Directory, 59
 - OpenLDAP, 69
 - infrastructure
 - assessing company's current, 2–8
 - authentication, 82
 - Linux. *See* Linux infrastructure
 - OpenLDAP, designing, 62–63
 - inheritance, dynamic and static, 111
 - inode described, 112
 - installing
 - AMANDA, 132–138
 - Apache, 264
 - Courier-IMAP server suite, 193, 227–228
 - Gnome and KDE desktops, 323
 - ISC BIND package, 42
 - Linux applications, 311–312
 - Linux on training computers, 304, 307
 - OpenLDAP, 64
 - printer drivers, 170–174
 - Samba, 84–88
 - smbldap tools, 88
 - Instant Messenger, 397
 - Integrated Development Environments (IDE), 295
 - Internet, .com domain names, 69
 - Internet Engineering Task Force website, 290
 - Internet Explorer and virus infections, 219
 - Internet Mail Access Protocol (IMAP), 192
 - Internet Service Providers. *See* ISPs
 - Internet System Consortium (ISC), 28
 - intrusions
 - described, 404–406
 - detection systems. *See* IDSs
 - inventorying servers, 5–6
 - IP addresses
 - assignment services, 23–31, 46–47
 - configuring for virtual hosting, 277–278
 - host names and, 31
 - testing DNS functionality, 45
 - ip link command, 392
 - IPSec (IP Security), 396
 - IPX (Internetwork Packet Exchange) and OSI model, 381
 - IRC (Internet Relay Chat), 397
 - ISC DHCP server, configuring, 41
 - ISPs (Internet Service Providers) and Internet e-mail, 206
 - offering anti-spam, anti-virus services, 224–225
 - sniffers and, 397
 - IT staff, training, migration assistance, 292–293
 - J**
 - Jackson, John R., 132
 - jitter, jabber, 369
 - journaling modes, Ext2/3 file system, 113
 - K**
 - KDE desktop, 309, 319, 321–323
 - KDE suite/KMail, 332–333
 - kdeprint command, 156
 - Kerberos authentication, 73
 - Kickstart installation profiler, 304
 - kiosks
 - assessing, 294, 298
 - and Linux Terminal Services model, 303
 - KMail, 330, 332–333
 - Knoppix security tools, 435
 - knowledge workers, assessing, 294–295
 - KOffice office suite, 355
 - Kolab groupware server, 333–334
 - Konqueror browser, 346
 - Kontakt software, 333
 - KWin window manager, 325
 - KWrite, 355
 - L**
 - LAMP (Linux-Apache-MySQL-PHP/Perl/Python)
 - architecture, 247
 - and Linux-based groupware, 236
 - LAN Manager (LANMAN), 57
 - LANMAN protocol, 74, 100
 - LDAP (Lightweight Directory Access Protocol)
 - authentication process, 75
 - configuration (ldap.conf), 78, 103
 - Exchange and, 183
 - function, directories, queries, 53–57
 - queries, understanding, 56–57
 - summary, 67–68
 - ldap.conf, 78, 103
 - ldapmin command, 153
 - libpcap packet-capture program, 401
 - LibPST e-mail conversion application, 339
 - licenses
 - General Public. *See* General Public Licenses (GPL)
 - Gnome desktop, 320
 - Lightweight Directory Access Protocol. *See* LDAP
 - Linux
 - authentication, 74–75, 101–102
 - clients. *See* Linux clients
 - common application file formats (table), 299
 - desktops. *See* Linux desktops
 - file systems, 145
 - migrating e-mail to, 228–229
 - migrating file services to, 140–144
 - migrating information from Exchange, 225–229
 - NTP clients, 35–36
 - Office application suites, 350–356, 361
 - PAM. *See* PAM
 - print services. *See* Linux print services
 - running Windows applications on, 356–359, 361–362
 - SCP (Secure Copy command), 122
 - virtual desktop, console, 311
 - Web browsers for, 342–350
 - Linux-Apache-MySQL-PHP/Perl/Python. *See* LAMP
 - Linux-based directory services
 - designing, 61–66, 69
 - Linux-based DNS services, 22
 - Linux-based file services, designing, 139–140
 - Linux-based groupware and calendaring services, 238–242, 246
 - Linux-based messaging services
 - components, 189–196
 - creating server diagram, 211–212
 - designing, 205–212, 232
 - e-mail, sending and receiving, 187–189
 - infrastructure. *See* Linux infrastructure
 - integrating anti-spam, anti-virus services, 212–225
 - Mail Access Agent (MAA), 192–193
 - Mail Delivery Agent, Mailstore, 195–196
 - Mail Transfer Agent (MTA), 193–194
 - Mail User Agent (MUA), 190–191
 - system components, mail user agents, 189–191
- Linux-BBC, 436
- Linux BIND/DHCP, migrating MS-DNS/DHCP to, 37–45
- Linux clients
 - and AMANDA installation, 137–138

- authentication settings, 77–78
- Linux desktops
 - building training computers, 304
 - common environments, 318–323
 - deploying, 312–313
 - designing, 301–304, 314
 - migrating to, generally, 292–294
 - testing, 305
 - training users, 307–312
- Linux DNS services, migrating
 - from Microsoft DNS
 - services to, 44–45
- Linux Documentation Project, 290, 308
- Linux infrastructure
 - designing, 11–13
 - determining requirements for, 8–11
 - migrating to, 16–17
 - testing, deploying, 13–16
- Linux NTFS, 111–112
- Linux print services
 - configuring using CUPS, 155–161
 - configuring with BSD or SVSV, 153–155
 - described, using, 151–153, 176
- Linux Terminal Services, using, 303
- Linux User Groups (LUGs), 290
- load balancing
 - directory services, 64
 - Exchange 5.5 support, 186
- local delivery agent (LDA), 195
- log files
 - Apache, 289
 - Courier e-mail, 204
 - IDS monitoring, 438
 - security, IDS parsing, 423
 - sniffer, 398
- logon process, Windows
 - 98/NT/2000/XP, 73–74
- lpc, lpr commands, 154

M

- MAC addresses
 - and Ethernet, 379
 - spoofing, 390, 421
- Macromedia Flash, 348
- macros in OpenOffice, 353–354
- Mail Access Agent (MAA), 192–193
- mail clients, 190
- Mail Transfer Agent (MTA)
 - development of, 197
 - Linux-based messaging services, 193–194
- maildir mail storage format, 196, 198, 228–229
- Maildrop, 195
- MailScanner virus scanner, 221
- managing
 - migration scheduling, 17
 - permissions, 118–123, 145–146
 - printers, 156–157

- mapping
 - URL paths to file system parts, 271
 - users to mailbox, 89
 - Windows SID to UNIX UID, 141
- MAPS RBL (Realtime Blackhole List), 216
- Master File Table (MST), 109
- mbx mail storage format, 195–196
- McCool, Rob, 263
- media, choosing backup, 125–126
- messaging server software
 - choosing, 208–209
 - open source, 197–204
- messaging services
 - generally, 182–183
 - Microsoft, 183–186
- Metacity window manager, 325
- Microsoft Certificate Services, 259
- Microsoft CHAP, 344
- Microsoft directory services, 57–58, 68
- Microsoft Internet Information Server. *See* IIS
- Microsoft messaging services, 183–186, 231
- Microsoft Office application suites, 350–356
- Microsoft Security Baseline Scanner, 470
- migrating
 - application data, profiles, 305–306
 - bookmarks, 347
 - constraints and adjustments (table), 10–11
 - e-mail/PIM (Personal Information Manager), 244
 - e-mail to Linux, 228–229, 337–342
 - Exchange information to Linux, 225–229
 - file services to Linux, 140–144
 - groupware and calendaring services, 246
 - to Linux groupware, calendaring services, 242–244
 - to Linux infrastructure, 16–17, 19
 - MS-DNS/DHCP to Linux BIND/DHCPD, 37–45, 48
 - from NT/Exchange, 88–93
 - planning summary, 18
 - static sites from IIS to Apache, 286, 287
 - Web services from IIS to Apache, 250–251
 - Windows logon authentication files, 97
 - Windows print services to CUPS/Samba, 175, 177–178

- Windows users to Linux desktop
 - generally, 292–294
- Minix file system, 113
- Mixed Mode Active Directory, 93
- Modern Microsoft Operating System (OS), 106
- Mondo backups, 129
- monitor resolution, changing, 310
- monitoring
 - Apache status, 264–265
 - company policy with IDS, 446
 - DHCP traffic, 28
 - hosts, 394
 - hub ports to detect sniffing, 393
 - intrusions. *See* IDSs (Intrusion Detection Systems)
 - network traffic by Carnivore, 375–378
 - systems calls, 423
 - mount points, volume, 110
- Mozilla, 298, 299, 307
 - importing Outlook mail into, 338–339
- Mail/Thunderbird, 334–335
 - suite described, 343–344
- mt command, 128
- multi-master replication, 60
- MyDoom worm, 445

N

- name resolution
 - DNS (Domain Name Service). *See* DNS
 - file-based, 31–32
- name resolution services, 47
- Name Service Switch (NSS), configuration, 77
- naming services
 - DNS. *See* DNS (Domain Name Service)
 - WINS (Windows Internet Naming Service), 34
- Naming System Venture, 115
- Native Mode Active Directory, 94, 103
- Nautilus, 307
- Nelson, Ted, 251
- Neped sniffing detector, 394
- Nessus Security Scanner
 - described, 407
 - enumerating services, 465
 - identifying services, applications, vulnerabilities, 466–468
 - and Microsoft vulnerabilities, 472
 - reports, 457
 - scan options, 463
- net rcp share command, 141
- Netscape Communications Corporation, 251
- network analysis
 - and company policy, 397
 - described, using, 366–378
 - port mirroring, 388
 - summary, 399

- network analyzers
 - common, described, 373–375
 - Ethereal Network Analyzer, 367
- Network Associates sniffer, 373
- network IDSs (NIDSs), 412–414, 443
- Network Information Services (NIS) authentication, 76
- network logon, Samba network, 86–87
- Network Monitor, 374
- network node IDS, 450
- network scanning tools, 407
- network services, guidelines for application configuration, 13
- Network Time Protocol (NTP), 34–35, 37
- networks
 - assessing vulnerabilities, 461
 - speed, and file services design, 140
- Newsham, Tim, 442
- NIDSs (network IDSs), 412–414
- Nimda worm, 408
- NIS+ client for Linux, 76
- Nmap scanner, 407, 449
- 'No Access' permissions, 109
- Novell GroupWise, OpenExchange servers, 332
- nsswitch.conf, 77
- NT Security Accounts Manager, 57–58
- NTFS file systems
 - dumping permissions to file, 143
 - Linux, 111–112
 - NTSF 4.0 vs. FAT, 110
 - understanding, 109–112
- NTFSPROGS v1.8.0 utilities, 112
- NTLM (NT LANMAN), 57
- NTLM authentication, 74, 84–88
- NTP (Network Time Protocol), 34–35, 37
- ntp.org, 35–36
- NTS exploits, 415
- O**
- Object Identifier (OIDs), 55
- objectclasses, 54, 65–66
- Office application suites, 350–356
- Offline NT Password & Registry Editor, 436
- OIDs (Object Identifier), 55
- Oliva, Alexandre, 132
- open source
 - Freshmeat.net repository, 301
 - groupware features, 247
 - messaging server software, 197–204, 209
- OPEN-XCHANGE (OX)
 - groupware, calendaring, 240–241
- OpenLDAP
 - configuring and testing servers, 64–66
 - directory solution described, 59
 - MUA support, 191
 - server daemons, utilities, 60–61
 - summary, 68–69
 - using for cross-platform authentication, 82
- OpenOffice.org Linux office suite, 351–354
- Opera browser, 346–347
- operating systems
 - backing up, 124
 - host file locations on various (table), 32
 - security, 434
- OSI (Open Systems Interconnection) model, 380–383
- Outlook
 - e-mail and PIMs, 329–330
 - groupware features, calendaring, 236–237, 245–246
 - migrating mail from, 337–338
 - Outport (Outlook migration application), 243–244
- Outlook Express and virus infections, 219
- Outport (Outlook migration application), 243–244, 339–341
- outsourcing anti-spam, anti-virus services, 224–225
- OWA (Outlook Web Access), 186
- P**
- PAC (Proxy Auto Configuration), Kerberos, 73
- packet sniffing, IDS function, 422–423
- packets, and network traffic, 381
- Packetalyzer sniffer, 375
- PAM (Pluggable Authentication Modules)
 - and OpenLDAP, 122
 - and user authentication, 79–81
- partitions, FAT, 106–109
- passwords
 - and authentication, 72, 76
 - and bind operations, 57
 - Offline NT Password & Registry Editor, 436
 - one-time, 396, 402
 - strong, 82
 - and Windows NT SAM, 57
- pathnames, determining for commands, 155
- PATRIOT Act, 414
- PCI cards, 140
- PDC (Primary Domain Controller), 57
- PDF files, OpenOffice limitations, 353–354
- penetration tests and vulnerability assessment, 479
- Peripheral Component Interconnect (PCI) card, 140
- permissions
 - See also* access control
 - Apache, 272–274
 - granting user, 109–110
 - managing, 118–123, 145–146
 - migrating file, 142–143
 - migrating share, 141
- PGP (Pretty Good Privacy), 396–397
- PHP / PEAR (PHP Extension and Application Repository), 239, 289
- PHP scripting language, 419
- PIM (Personal Information Manager)
 - choosing software, 336–337
 - clients, and e-mail, 329–330, 360
 - e-mail, and migration methods, 244
- planning
 - migration of desktop users, 292–294
 - for testing, 14–16
 - for training in new system, 17
- plug-ins
 - groupware, 247
 - Web browser, 347–350
- policies
 - Acceptable Use, 414
 - network analysis and, 397
 - using IDS to monitor company's, 446
- POP (Post Office Protocol), 192
- port mirroring and network analysis, 388
- port scanning, 406
- port spanning, 388, 391
- Posix accounts and Webmin modules, 62
- POSIX ACLs, 118
- post-migration infrastructure design diagrams
 - Acme Widgets, 12
 - Ballystyx Engineering, 13
- Post Office Protocol (POP), 192
- Postfix messaging software, 199–200
- PostScript printing, 150–151, 154, 156, 174
- PPP (Point-to-Point Protocol) and OSI model, 381
- Pretty Good Privacy (PGP), 396–397
- Primary Domain Controller (PDC), 57
- Primary Windows NT Account (Assoc-NT-Account), 58
- print servers, vulnerability assessment, 474

- print services
 - CUPS. *See* CUPS
 - generally, 150, 176
 - Linux. *See* Linux print services
 - Windows. *See* Windows print services
- printer drivers, 150, 159, 170–174
- Printer Job Language (PJL), 151
- printing
 - commands (tables)
 - PostScript, 150–151, 154, 156, 174
 - style defaults for various systems (table), 165–166
- profiles
 - migrating to Linux desktop, 305–306
 - roaming, 86–87, 97
- PromiScan Ver. 0.27, 394
- Promisc.c program, 395
- PromiscDetect, 392
- protocol analysis and intrusion detection, 451
- protocols
 - See also specific protocol*
 - broadcast, 378
 - Exchange 5.5 support (table), 185
- Ptacek, Tom, 442
- PTR (pointer) resource records, 32
- Q**
 - Qmail, 197, 197–199, 234
 - queries, LDAP, 55–57
- R**
 - RAT (Remote Admin Trojan), 371
 - RealPlayer plug-in, 348–349, 362
 - RedHat Linux
 - Courier-IMAP server suite, 227
 - and Fedora, 263
 - graphical administration tools, 284–285
 - Kickstart installation profiler, 304
 - regback.exe, regrest.exe, 130
 - Reiser, Hans, 115
 - ReiserFS file system, 112, 115–117
 - remote administration
 - of CUPS, 178
 - and Windows applications, 356
 - remote procedure calls (RPCs), 185, 381
 - remote servers, obtaining list of existing shares, 141
 - Renfro, Chad, 383
 - reparse points, 110
 - replication
 - with OpenLDAP, 60
 - options, 123–129, 146
 - Request For Proposal (RFP), 300
 - restoring
 - ACLs, 122
 - options, 123–129, 146
 - Return On Investment (ROI)
 - IDSs (Intrusion Detection Systems), 449–450
 - of Windows to Linux desktop migration, 316
 - RFC (Request for Comments), 289–290
 - RFP (Request For Proposal), 300
 - roaming profiles, 86–87, 97
 - rogue DHCP servers, 30
 - root servers, 32
 - rootkits, 371–372
 - rotating backup media, 127–129
 - RPCs (remote procedure calls), 185, 381
- S**
 - Safari browser, 346
 - SAM (Security Accounts Manager), 57–58
 - Samba
 - accounts and Webmin modules, 62
 - installing, configuring, 84–88
 - migrating Windows print services with CUPS, 175, 177–178
 - printers, 161–170, 177, 179
 - replacing NT4 DC, Active Directory server, 147
 - servers, configuring
 - authentication components, 88–89
 - use- and group-based controls (table), 123
 - Samba Web Administration Tool (SWAT), 161
 - samba.schema, 65
 - SAN (Storage Area Network), 140, 186
 - SASL (Simple Authentication and Security Layer), 75
 - SATAN (Security Administrator Tool for Analyzing Networks), 461
 - Sawfish window manager, 325
 - scalability of messaging services, 205
 - scanning networks, 407
 - scheduling
 - backups, 126–127
 - migration, managing, 17
 - schemas
 - directory, 55
 - samba.schema, 65
 - scope, determining DHCP, and options on Windows NT, 38–39
 - script kiddies, 431
 - scriptaliases (Apache), 271–272
 - scripts
 - smbldap, 88
 - w2lmt-migrate-smbauth, 90
 - SCSI (Small Computer Systems Interface) and clustered Exchange 5.5, 186
 - Secure Shell (SSH), 396
 - security
 - attacks. *See* attacks
 - controlling Apache, 272–274
 - in IIS, 257–258
 - NTP traffic restrictions, 27
 - strong passwords and, 82
 - vulnerability assessment. *See* vulnerability assessment
 - Security Administrator Tool for Analyzing Networks (SATAN), 461
 - Security Baseline Scanner (Microsoft), 470
 - Security Configuration Manager (SCM), 110
 - SecurityFocus Web site, 373
 - Sendmail, 197–198
 - Sentinel sniffing detector, 394
 - server certificates, 98, 278–281
 - server diagrams, creating Linux e-mail, 211–212
 - servers
 - DHCP. *See* DHCP servers
 - directory. *See* directory servers
 - graphical administration tools, 284–285
 - inventorying, 5–6, 18
 - placement in post-migration infrastructure, 12–13
 - print, 474
 - stratum 1, 35
 - terminal, 303
 - Web. *See* Web servers
 - setfacl command, 119
 - share permissions, 141–146
 - shares
 - obtaining list on remote servers, 141
 - print, 170–174
 - sharing Samba printers, 161–170, 177
 - Shavlik Technology's HFNetChk Pro, 472
 - Shockwave/Director, 348
 - SID (Security Identifier) and Windows NT SAM, 57
 - Simple NTP (SNTP), 34
 - Skolnick, Cliff, 263
 - slapd (Stand-alone LDAP Daemon), 60
 - slapd.conf, 89, 98
 - slave servers, 57
 - slurpd (Stand-alone Update Resolution Daemon), 60
 - SM (Security Configuration Manager), 110
 - Small Computer Systems Interface (SCSI)
 - and clustered Exchange 5.5, 186
 - tape drives for backups, 127–128
 - smb.conf
 - described, 85–86
 - printing parameters, variables, 161–170
 - smbldap tools, 88
 - SMTP (Simple Mail Transfer Protocol) and OSI model, 380

- sniff.c packet sniffer, 383–384
 - Sniffer (product line), 367
 - sniffers
 - common, 373–375
 - detecting, 391–395
 - protecting against, 395–397
 - writing your own, 383–384
 - sniffing
 - connections, other data, 372–373
 - described, using, 366–367
 - Ethernet and, 378–380
 - hubs, 387
 - intruder's use of, 370–372
 - log files, 398
 - summary, 399–400
 - Sniffit network sniffer, 374
 - SNMP (Simple Network Management Protocol) and OSI model, 380
 - Snoop network sniffer, 374
 - Snort network intrusion detection, 374
 - integrating with your security system, 444–446
 - as packet sniffer, 413
 - and SQL Slammer worm, 409
 - Snot program, and Snort, 448
 - SNTP (Simple NTP), 34
 - software, asset lists, 296–299
 - SolarWinds scanner, 407, 445
 - Sourceforge
 - groupware, 236
 - open source applications, 316
 - spam
 - described, analyzing, fighting, 213–216
 - fingerprinting, 214–215
 - SpamAssassin, 214, 219–220
 - Span (Switched Port ANalyzer), 389
 - SPF (Sender Policy Framework), 213
 - spoofing MAC addresses, 390
 - SQL Slammer worm, 408–409
 - SQL Snake worm, 432
 - SquirrelMail, 190
 - SqWebMail, 190, 202–203
 - srvcheck utility, 143
 - SSH (Secure Shell), 396
 - SSL (Secure Sockets Layer), 280, 290, 396
 - SSL/TLS (Secure Sockets Layer/Transport Layer Security), 258–259
 - star archives, 122
 - Star Writer, Impress, Calc, Web applications, 351–354
 - StarOffice office suite, 354–355
 - Start of Authority (SOA) record, 33
 - starting
 - Apache, 264–265
 - Samba services, 87
 - Static and Dynamic IP address configurations (table), 23–24
 - stealth mode, intrusion detection, 418
 - Stick program, and Snort, 448
 - stopping
 - Apache, 264–265
 - Samba services, 87–88
 - storage, e-mail, 210
 - Storage Area Network (SAN), 140, 186
 - StoreBackup, 129
 - stratum 1 servers, 35
 - strong passwords, 82
 - su command, 154
 - SublinAd tool, 144
 - SubSeven sniffer, 371
 - Sun Microsystems
 - NIC service, 77
 - PAM, 79
 - switches
 - defeating, 389–391
 - described, using, 386–387
 - flooding programs, 401
 - Sylpheed e-mail application, 335–336
 - system administrators
 - approach to vulnerability assessment, 469–470
 - managing migration to Linux infrastructure, 17
 - system call monitoring, IDSs (Intrusion Detection Systems), 423
 - System V (SYSV)-based printing system. *See* SYSV
- ## T
- T0rnKit rootkit, 371
 - Tab Window Manager (TWM), 325
 - tape drives, setting up AMANDA backup, 138–139
 - tar archives, 122, 128
 - Tcddump, 374
 - TCP/IP (Transmission Control Protocol/Internet Protocol)
 - and HTML, 251
 - IP address assignment services, 23–31
 - migrating MS-DNS/DHCP to Linux BIND/DHCPD, 37–45
 - networking services, introduction, 22–23
 - routing protocol described, 382–383
 - time synchronization services, 34–37
 - understanding name resolution services, 31–34
 - tcpdump, analyzing DHCP traffic with, 28
 - technical support for Linux, 313
 - technical workers, assessing, 295, 298
 - templates, configuring DHCP server from, 28
 - test labs, using, 16
 - test plans, 14–16, 19
 - testing
 - authentication services, 97, 99
 - e-mail functionality, 14
 - hacker penetration, 397
 - Linux desktop, 305, 315
 - Linux infrastructure, 13–16, 19
 - OpenLDAP servers, 64–66
 - PHP, 289
 - stage of desktop migration, 293
 - Thunderbird e-mail application, 335
 - time synchronization services, 34–37, 47
 - TLS/SSL (Transport Layer Security/Secure Sockets Layer) and LDAP authentication, 75, 83
 - token types in configuration files, 79–80
 - tokens and Bayesian analysis, 215
 - trade secrets, and security intrusions, 406
 - training
 - computers, building, 304
 - and desktop migration roadmap, 292–293
 - desktop users, 307–312
 - planning for, 17
 - transactional workers
 - assessing, 295
 - and Linux Terminal Services model, 303
 - Transmission Control Protocol/Internet Protocol. *See* TCP/IP
 - Transport Layer Security/Secure Sockets Layer (TLS/SSL), 75, 83
 - Transport Layer Security (TLS), 258
 - Tripwire tool, 424
 - Trojan horse attacks, 216, 434
 - troubleshooting
 - Samba server setup, 147–148
 - testing Web server operation, PHP, 289
 - Web services, 290
 - TWiki knowledge management systems, 241–242
- ## U
- UDP port 123, 37–45
 - UDP (User Datagram Protocol) and SQL Slammer worm, 408
 - 'Under Construction' page, 253
 - Uniform Resource Indicator (URI), representing LDAP search, 57
 - UNIX
 - Common UNIX Print System. *See* CUPS
 - Linux. *See* Linux permissions, 118

- print services, 152
- U.S. Economic Espionage Act of 1996, 406
- usage reports Courier e-mail, 204
- user accounts, passwords and, 57
- user permissions, managing, 109–110, 118–123
- user profiles, importing to Linux desktop, 307
- user types, assessing, 294–295
- usernames, managing authentication with, 76
- users
 - migrating to Linux infrastructure, 16–17
 - training for Linux desktop, 307–312

V

- vCard calendaring file format, 244
- Vcard standard, 342
- .vcf files, 244
- VFAT (Virtual File Allocation Table), 109
- virtual desktop, console, 311
- virtual directories (IIS), 256
- Virtual File Allocation Table (VFAT), 108
- virtual hosting, Apache, 276–282
- virtual private networks (VPNs), 395, 402
- virtual servers, configuring IIS, 260–262
- viruses
 - anti-spam, anti-virus applications, 219–222
 - described generally, 216
 - and intrusion detection, 408–409
 - protecting against, 232–233
 - Snort and, 445
 - types of payloads (table), 217–218
- Visio diagramming program, 6
- VPNs (virtual private networks), 395, 402
- vulnerabilities, reporting, 468
- vulnerability assessment
 - administrative, outsider, and hybrid approaches, 469–473
 - automated, 461–468
 - and IDSs, 479–480
 - introduction to, 456–459
 - limitations of, 473–475
 - summary, 476–478
 - types of assessments, 459–461

W

- w2lmt-migrate-smbauth configuration file, 93–97
- WANs (wide area networks), locating home directory for file services, 139

- Web browsers
 - for Linux systems, 342–350
 - plug-ins, 347–350
- Web servers
 - Apache. *See* Apache Web Server
 - Apache virtual hosting, 276–282
 - IIS (Internet Information Server). *See* IIS
 - uses of, and HTTP, 250–251
- Web services
 - introduction to, 250–251
 - migrating from IIS to Apache, 250–251
- Webadmin (Courier), 203–204
- Webmin tool, 49
- WebMin Web administration tool, 263, 285
- Websites
 - available sniffers, 373
 - CMU's Cyrus SASL implementation, 75
 - eGroupware, 239
 - Freshmeat.net, 301
 - Hancom Office suite, 356
 - IIS default, 255
 - Internet Engineering Task Force, 290
 - Naming System Venture, 115
 - OpenSSL, 290
 - PAM, 81
 - spam fingerprinting organizations, 215
 - Wikis, 241–242
 - X.org, 325
- whereis command, 154
- whitelisting spammers, 216
- Wikis, TWiki flavor of, 242
- WikiWikiWebs control management system, 241–242
- Williams, Adam, 53
- winbindd, 141
- window managers, 311, 323, 324, 359–360
- WindowMaker window manager, 326
- Windows
 - applications, Linux equivalents, 311–312
 - applications, running on Linux, 356–359, 361–362
 - common file formats (table), 299
 - desktop differences from Linux, 309–312
- Windows 2000
 - authentication generally, 73
 - DHCP Manager, 39–40
 - Network Monitor, 374
- Windows 98
 - authentication generally, 73–74
 - configuring BIND and DHCP for DDNS, 33–34

- Windows clients and AMANDA installation, 138
- Windows file systems
 - File Allocation Table (FAT), 106–109
 - generally, 145
 - NTFS file systems, 109–112
- Windows Internet Naming Service (WINS), 34
- Windows NT
 - authentication generally, 73–74
 - configuring BIND and DHCP for, 33–34
 - determining DHCP scopes, options on, 38–39
 - domains, 58, 59
 - migrating, 88–93
 - Network Monitor, 374
 - Security Accounts Manager (SAM), 57–58
- Windows NT Resource Kit, backup and restore utilities, 130
- Windows print services, using, 150–151, 176
- Windows time service clients, 36–37
- WinDump network analyzer, 373
- WINE native Windows applications, 302
- wine some.exe command, 302
- Wine tool, 357–358, 361
- WINS (Windows Internet Naming Service), 34
- wiretaps, and intrusion detection, 405
- World Wide Web (WWW) and HTTP, 250–251
- worms
 - See also specific worms*
 - and intrusion detection, 408–409
 - Snort and, 445
- WYSIWYG printing, 150

X

- X Window system and window managers, 324–327, 359
- Xcals.exe, 143
- xcopy command, 141
- Xfe desktop environment, 323
- xmt command, 128

Z

- zone files, 33
- zone transfers, DNS, 43–44

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any

change.

- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program

by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) *year name of author*

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details

type 'show w'. This is free software, and you are welcome

to redistribute it under certain conditions; type 'show c'

for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright

interest in the program 'Gnomovision'

(which makes passes at compilers) written

by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

SYNGRESS PUBLISHING LICENSE AGREEMENT

THIS PRODUCT (THE “PRODUCT”) CONTAINS PROPRIETARY SOFTWARE, DATA AND INFORMATION (INCLUDING DOCUMENTATION) OWNED BY SYNGRESS PUBLISHING, INC. (“SYNGRESS”) AND ITS LICENSORS. YOUR RIGHT TO USE THE PRODUCT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT.

LICENSE: Throughout this License Agreement, “you” shall mean either the individual or the entity whose agent opens this package. You are granted a limited, non-exclusive and non-transferable license to use the Product subject to the following terms:

(i) If you have licensed a single user version of the Product, the Product may only be used on a single computer (i.e., a single CPU). If you licensed and paid the fee applicable to a local area network or wide area network version of the Product, you are subject to the terms of the following subparagraph (ii).

(ii) If you have licensed a local area network version, you may use the Product on unlimited workstations located in one single building selected by you that is served by such local area network. If you have licensed a wide area network version, you may use the Product on unlimited workstations located in multiple buildings on the same site selected by you that is

served by such wide area network; provided, however, that any building will not be considered located in the same site if it is more than five (5) miles away from any building included in such site. In addition, you may only use a local area or wide area network version of the Product on one single server. If you wish to use the Product on more than one server, you must obtain written authorization from Syngress and pay additional fees.

(iii) You may make one copy of the Product for back-up purposes only and you must maintain an accurate record as to the location of the back-up at all times.

PROPRIETARY RIGHTS; RESTRICTIONS ON USE AND TRANSFER: All rights (including patent and copyright) in and to the Product are owned by Syngress and its licensors. You are the owner of the enclosed disc on which the Product is recorded. You may not use, copy, decompile, disassemble, reverse engineer, modify, reproduce, create derivative works, transmit, distribute, sublicense, store in a database or retrieval system of any kind, rent or transfer the Product, or any portion thereof, in any form or by any means (including electronically or otherwise) except as expressly provided for in this License Agreement. You must reproduce the copyright notices, trademark notices, legends and logos of Syngress and its licensors that appear on the Product on the back-up copy of the Product which you are permitted to make hereunder. All rights in the Product not expressly granted herein are reserved by Syngress and its licensors.

TERM: This License Agreement is effective until terminated. It will terminate if you fail to comply with any term or condition of this License Agreement. Upon termination, you are obligated to return to Syngress the Product together with all copies thereof and to purge and destroy all copies of the Product included in any and all systems, servers and facilities.

DISCLAIMER OF WARRANTY: THE PRODUCT AND THE BACK-UP COPY OF THE PRODUCT ARE LICENSED "AS IS". SYNGRESS, ITS LICENSORS AND THE AUTHORS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO RESULTS TO BE OBTAINED BY ANY PERSON OR ENTITY FROM USE OF THE PRODUCT AND/OR ANY INFORMATION OR DATA INCLUDED THEREIN. SYNGRESS, ITS LICENSORS AND THE AUTHORS MAKE NO EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE WITH RESPECT TO THE PRODUCT AND/OR ANY INFORMATION OR DATA INCLUDED THEREIN. IN ADDITION, SYNGRESS, ITS LICENSORS AND THE AUTHORS MAKE NO WARRANTY REGARDING THE ACCURACY, ADEQUACY OR COMPLETENESS OF THE PRODUCT AND/OR ANY INFORMATION OR DATA INCLUDED THEREIN. NEITHER SYNGRESS, ANY OF ITS LICENSORS, NOR THE AUTHORS WARRANT THAT THE FUNCTIONS CONTAINED IN THE PRODUCT WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE PRODUCT WILL BE UNINTERRUPTED OR ERROR FREE. YOU ASSUME THE ENTIRE RISK WITH RESPECT TO THE QUALITY AND PERFORMANCE OF THE PRODUCT.

LIMITED WARRANTY FOR DISC: To the original licensee only, Syngress warrants that the enclosed disc on which the Product is recorded is free from defects in materials and workmanship under normal use and service for a period of ninety (90) days from the date of purchase. In the event of a defect in the disc covered by the foregoing warranty, Syngress will replace the disc.

LIMITATION OF LIABILITY: NEITHER SYNGRESS, ITS LICENSORS NOR THE AUTHORS SHALL BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE, CONSEQUENTIAL OR SIMILAR DAMAGES, SUCH AS BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS, RESULTING FROM THE USE OR INABILITY TO USE THE PRODUCT EVEN IF ANY OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL APPLY TO ANY CLAIM OR CAUSE WHATSOEVER WHETHER SUCH CLAIM OR CAUSE ARISES IN CONTRACT, TORT, OR OTHERWISE. Some states do not allow the exclusion or limitation of indirect, special or consequential damages, so the above limitation may not apply to you.

U.S. GOVERNMENT RESTRICTED RIGHTS. If the Product is acquired by or for the U.S. Government then it is provided with Restricted Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in FAR 52.227-19. The contractor/manufacturer is Syngress Publishing, Inc. at 800 Hingham Street, Rockland, MA 02370.

GENERAL: This License Agreement constitutes the entire agreement between the parties relating to the Product. The terms of any Purchase Order shall have no effect on the terms of this License Agreement. Failure of Syngress to insist at any time on strict compliance with this License Agreement shall not constitute a waiver of any rights under this License Agreement. This License Agreement shall be construed and governed in accordance with the laws of the Commonwealth of Massachusetts. If any provision of this License Agreement is held to be contrary to law, that provision will be enforced to the maximum extent permissible and the remaining provisions will remain in full force and effect.

***If you do not agree, please return this product to the place of purchase for a refund.**

Syngress: The Definition of a Serious Security Library

Syn-gress (sin-gres): *noun, sing.* Freedom from risk or danger; safety. See *security*.



AVAILABLE NOW
order @
www.syngress.com

Snort 2.1 Intrusion Detection, Second Edition

Jay Beale, Brian Caswell, et. al.

"The authors of this *Snort 2.1 Intrusion Detection, Second Edition* have produced a book with a simple focus, to teach you how to use Snort, from the basics of getting started to advanced rule configuration, they cover all aspects of using Snort, including basic installation, preprocessor configuration, and optimization of your Snort system."

—Stephen Northcutt

Director of Training & Certification, The SANS Institute

ISBN: 1-931836-04-3

Price: \$49.95 U.S. \$69.95 CAN

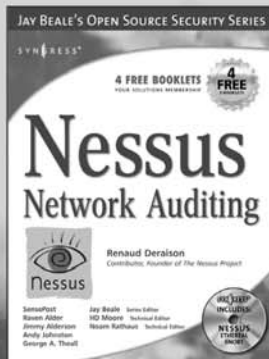
Ethereal Packet Sniffing

Ethereal offers more protocol decoding and reassembly than any free sniffer out there and ranks well among the commercial tools. You've all used tools like tcpdump or windump to examine individual packets, but Ethereal makes it easier to make sense of a stream of ongoing network communications. Ethereal not only makes network troubleshooting work far easier, but also aids greatly in network forensics, the art of finding and examining an attack, by giving a better "big picture" view. *Ethereal Packet Sniffing* will show you how to make the most out of your use of Ethereal.

ISBN: 1-932266-82-8

Price: \$49.95 U.S. \$77.95 CAN

AVAILABLE NOW
order @
www.syngress.com



AVAILABLE NOW
order @
www.syngress.com

Nessus Network Auditing

Jay Beale, Haroon Meer, Roelof Temmingh, Charl Van Der Walt, Renaud Deraison

Crackers constantly probe machines looking for both old and new vulnerabilities. In order to avoid becoming a casualty of a casual cracker, savvy sys admins audit their own machines before they're probed by hostile outsiders (or even hostile insiders). Nessus is the premier Open Source vulnerability assessment tool, and was recently voted the "most popular" open source security tool of any kind. *Nessus Network Auditing* is the first book available on Nessus and it is written by the world's premier Nessus developers led by the creator of Nessus, Renaud Deraison.

ISBN: 1-931836-08-6

Price: \$49.95 U.S. \$69.95 CAN